

# Off-centre Steiner points for Delaunay-refinement on curved surfaces<sup>☆</sup>

Darren Engwirda<sup>a,\*\*</sup>, David Ivers<sup>a</sup>

<sup>a</sup>*School of Mathematics and Statistics F07, University of Sydney, NSW 2006, Australia*

## Abstract

An extension of the restricted Delaunay-refinement algorithm for surface mesh generation is described, where a new point-placement scheme is introduced to improve element quality in the presence of mesh size constraints. Specifically, it is shown that the use of *off-centre* Steiner points, positioned on the faces of the associated Voronoi diagram, typically leads to significant improvements in the shape- and size-quality of the resulting surface tessellations. The new algorithm can be viewed as a Frontal-Delaunay approach – a hybridisation of conventional Delaunay-refinement and advancing-front techniques in which new vertices are positioned to satisfy both element size and shape constraints. The performance of the new scheme is investigated experimentally via a series of comparative studies that contrast its performance with that of a typical Delaunay-refinement technique. It is shown that the new method inherits many of the best features of classical Delaunay-refinement and advancing-front type methods, leading to the construction of smooth, high quality surface triangulations with bounded radius-edge ratios and convergence guarantees. Experiments are conducted using a range of complex benchmarks, verifying the robustness and practical performance of the proposed scheme.

**Keywords:** Surface mesh generation, Delaunay-refinement, Advancing-front, Frontal-Delaunay, Off-centre Steiner points, Element quality

## 1. Introduction

Surface mesh generation is a key component in a variety of computational modelling and simulation tasks, including many forms of computational engineering and numerical modelling, a range of problems in computer graphics and animation, and various data visualisation applications. Given a geometric domain described by a bounding surface  $\Sigma$  embedded in  $\mathbb{R}^3$ , the surface *triangulation* problem consists of tessellating  $\Sigma$  into a *mesh* of non-overlapping triangular elements, such that all geometrical, topological and user-defined constraints are satisfied. While each use-case contributes its own set of specific considerations, it is typical to require that surface tessellations: (i) consist of elements of high shape-quality, (ii) provide good geometrical and topological approximations to the underlying surface  $\Sigma$ , and (iii) satisfy a set of user-specified element sizing constraints. While various strategies have been developed to solve the surface meshing problem in the past, a new algorithm is developed in this study with the aim of improving the *quality* of the resulting triangulations.

### 1.1. Related Work

Mesh generation is a broad and evolving area of research. A number of algorithms have been developed to tackle various meshing tasks, including *grid-overlay* methods [2, 3] in which a mesh is generated by warping and refining a semi-structured background grid, *advancing-front* techniques [4, 5, 6, 7] which grow a mesh incrementally in a layer-wise fashion, and *Delaunay-based* strategies [8, 9, 10, 11, 12, 13, 14, 15, 16, 17] which incrementally refine an initially coarse Delaunay triangulation.

The Delaunay triangulation [18] is imbued with a number of attractive theoretical properties. In the context of mesh generation, for triangulations in  $\mathbb{R}^2$  and surface tessellations embedded in  $\mathbb{R}^3$ , it is known that such structures are topologically optimal [19], maximising the worst-case element quality in the resulting mesh. The reader is referred to [19] for additional details. As the name suggests, Delaunay-refinement schemes are based on the incremental *refinement* of an initially coarse bounding Delaunay tessellation. At each step of the algorithm, elements that violate a set of geometrical, topological or user-defined constraints are identified and the worst offending element is *eliminated*. This is achieved through the insertion of an additional *Steiner* vertex at the refinement point of the element – either the circumcentre of the element itself, or a point on an adjacent segment of the bounding geometry. The original planar Delaunay-refinement methods of Chew [8] and Ruppert [9, 10] have since been extended to support volumetric tessellations [12] and surface triangulations [15, 16, 14].

<sup>☆</sup>A short version of this paper appears in the proceedings of the 23rd International Meshing Roundtable [1].

\*Corresponding author. Tel.: +1-212-678-5573.

\*\*Current address: Department of Earth, Atmospheric and Planetary Sciences, Room 54-1517, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307.

Email addresses: d.engwirda@maths.usyd.edu.au; engwirda@mit.edu (Darren Engwirda), david.ivers@sydney.edu.au (David Ivers)

Delaunay-based methods have previously been applied to the surface meshing problem, via the so-called *restricted* Delaunay paradigm originally introduced by Edelsbrunner and Shah [20]. Given a surface  $\Sigma$  embedded in  $\mathbb{R}^3$  and a set of points  $X \in \Sigma$ , the restricted Delaunay surface triangulation  $\text{Del}|_{\Sigma}(X)$  is a sub-complex of the full-dimensional Delaunay tessellation  $\text{Del}(X)$ , containing the set of 2-faces  $f \in \text{Del}(X)$  that approximate the underlying surface. Boissonnat and Oudot [15, 16] have shown that when the point-wise sampling  $X$  is sufficiently *dense* and *well-distributed*, the restricted Delaunay triangulation  $\text{Del}|_{\Sigma}(X)$  is an accurate piecewise approximation of  $\Sigma$ , incorporating several theoretical guarantees of fidelity. Given such behaviour, a Delaunay-refinement algorithm for surface mesh generation proceeds as per the planar case, with an initially coarse restricted Delaunay triangulation, induced by a sparse sampling  $X \in \Sigma$ , being incrementally refined via the introduction of new Steiner vertices positioned on the surface  $\Sigma$ . This process is discussed in detail in Section 2.

Advancing-front techniques are an alternative approach to mesh generation, in which elements are created one-by-one, starting from a set of *frontal* facets initialised on the bounding geometry. The mesh is *marched* inwards layer-by-layer, with new elements created by carefully positioning vertices adjacent to facets in the frontal set. Following the placement of new elements, the set of frontal facets is updated to incorporate the new mesh topology. Elements are added incrementally in this fashion until a complete tessellation of the domain is obtained. Such methods are applicable to surface meshing problems [5, 6, 7], although several non-trivial issues of robustness are known to exist [7]. Fundamentally, advancing-front techniques are *heuristic* in nature, and do not typically offer theoretical guarantees of convergence or bounds on element quality. Nonetheless, when such methods are successful, they are known to produce very high-quality tessellations.

Due to their theoretical robustness, Delaunay-refinement schemes are a popular choice for practical meshing algorithms. On the other hand, it is well known that optimised advancing-front type methods can often produce meshes with superior element quality characteristics, though their heuristics can sometimes break-down in practice. In this study, a new *Frontal-Delaunay* algorithm is presented which seeks to combine the benefits of classical Delaunay-refinement and advancing-front type approaches. While such strategies have been pursued previously in the case of planar and/or volumetric refinement algorithms, it is believed that the present study is the first application of such ideas to the restricted surface meshing problem directly. The new algorithm is designed to produce smooth, high-quality triangulations consistent with an advancing-front scheme, whilst also inheriting the theoretical robustness of Delaunay-based techniques. It is expected that this new algorithm may be of interest to users who place a high premium on mesh quality, including those operating in the areas of computational engineering

and numerical simulation.

A brief review of the traditional surface Delaunay-refinement scheme is presented in Section 2, along with a discussion of the underlying theoretical constructs, including the *restricted* Delaunay tessellation. The new Frontal-Delaunay algorithm is presented in Section 3, focusing in-detail on the *off-centre* point-placement scheme proposed in this work. In Section 5, a detailed comparison between the conventional Delaunay-refinement and proposed Frontal-Delaunay schemes is presented, contrasting output quality and computational performance.

## 2. Restricted Delaunay-refinement Techniques

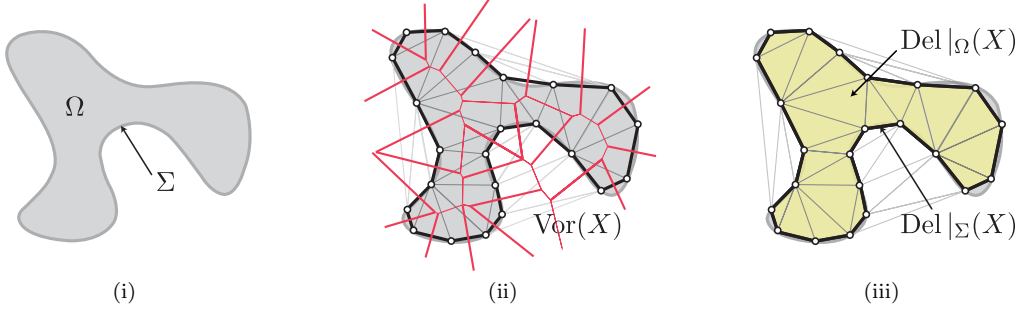
Delaunay-refinement algorithms operate by incrementally adding new Steiner vertices to an initially coarse restricted Delaunay triangulation of the underlying surface. Contrary to typical planar algorithms [8, 9, 10], refinement schemes for surface meshing are designed not only to ensure that the resulting mesh satisfies element shape and size constraints, but that the geometry and topology of the mesh itself is an accurate piecewise approximation of the underlying surface.

Before discussing the details of Delaunay-refinement algorithms for surface meshing problems, a number of important theoretical concepts are introduced:

**Definition 1.** Let  $\Sigma$  be a smooth surface embedded in  $\mathbb{R}^3$ , enclosing a volume  $\Omega \subset \mathbb{R}^3$ . Let  $\text{Del}(X)$  be a full-dimensional Delaunay tessellation of a point-wise sample  $X \subseteq \Sigma$  and  $\text{Vor}(X)$  be the Voronoi diagram associated with  $\text{Del}(X)$ . The *restricted Delaunay surface triangulation*  $\text{Del}|_{\Sigma}(X)$  is a sub-complex of  $\text{Del}(X)$  including any 2-face  $f \in \text{Del}(X)$  associated with an edge  $\mathbf{v}_f \in \text{Vor}(X)$  such that  $\mathbf{v}_f \cap \Sigma \neq \emptyset$ . The *restricted Delaunay volume triangulation*  $\text{Del}|_{\Omega}(X)$  is a sub-complex of  $\text{Del}(X)$  including any 3-simplex  $\tau \in \text{Del}(X)$  associated with an *internal* circumcentre  $\mathbf{c} \in \Omega$ .

It has been shown by a number of authors, including Amenta and Bern [21], Cheng, Dey, Levine [22], Cheng, Dey and Ramos [14], Cheng, Dey, Ramos and Ray [23], Boissonnat and Oudot [15, 16] and Cheng, Dey and Shewchuk [19], that given a *sufficiently dense* point-wise sampling of the surface,  $X \in \Sigma$ , the restricted Delaunay tessellation  $\text{Del}|_{\Sigma}(X)$  is guaranteed to be both geometrically and topologically representative of the underlying surface  $\Sigma$ . Specifically, it is known that, when the restricted tessellation is a so-called *loose  $\epsilon$ -sample* [15, 16],  $\text{Del}|_{\Sigma}(X)$  is homeomorphic to the surface  $\Sigma$ , the Hausdorff distance  $H(\text{Del}|_{\Sigma}(X), \Sigma)$  is small and that  $\text{Del}|_{\Sigma}(X)$  provides a good piecewise approximation of the geometrical properties of  $\Sigma$ , including its normals, area and curvature. Similar theoretical guarantees extend to the associated restricted volume tessellation  $\text{Del}|_{\Omega}(X)$ . The properties of restricted Delaunay tessellations are well documented in the literature, and a full account is not given here. The

Figure 1: Restricted tessellations for a curved domain in  $\mathbb{R}^2$ , showing (i) the curve  $\Sigma$  and enclosed area  $\Omega$ , (ii) the Delaunay tessellation  $\text{Del}(X)$  and Voronoi diagram  $\text{Vor}(X)$ , and (iii) the restricted curve and area tessellations  $\text{Del}|_{\Sigma}(X)$  and  $\text{Del}|_{\Omega}(X)$ .



reader is referred to the detailed expositions presented in [15, 16] and [19] for further details and proofs.

**Definition 2.** Let  $\text{Del}|_{\Sigma}(X)$  be a restricted Delaunay triangulation of a smooth surface  $\Sigma$ . Given a 2-simplex  $f \in \text{Del}|_{\Sigma}(X)$ , any circumball of  $f$  centred on the surface  $\Sigma$  is known as a *surface Delaunay ball* of  $f$ , denoted  $\text{SDB}(f)$ .

Surface Delaunay balls are centred at the intersection of the Voronoi diagram  $\text{Vor}(X)$  with the surface  $\Sigma$ . Specifically, each 2-face  $f \in \text{Del}|_{\Sigma}(X)$  is associated with an orthogonal edge in the Voronoi diagram  $\mathbf{v}_f \in \text{Vor}(X)$ , which is guaranteed, by definition, to intersect with the surface  $\Sigma$  at least once. In some cases, especially when the sampling  $X \in \Sigma$  is relatively coarse, there may be multiple surface intersections associated with a given 2-face  $f \in \text{Del}|_{\Sigma}(X)$ . In such cases, it is typical to consider only the largest associated surface ball. See Figure 2 for an illustration of the surface Delaunay ball for a general facet  $f$ .

**Definition 3.** Let  $\text{Del}|_{\Sigma}(X)$  be a restricted Delaunay surface triangulation of a smooth surface  $\Sigma$ . Given a 2-simplex  $f \in \text{Del}|_{\Sigma}(X)$ , the *surface discretisation error*  $\epsilon(f)$  is the Euclidean distance between the centres of the largest surface Delaunay ball of  $f$  and its diametric ball.

The surface discretisation error is a measure of how well the restricted triangulation  $\text{Del}|_{\Sigma}(X)$  approximates the underlying surface  $\Sigma$  geometrically. Considering a 2-face  $f \in \text{Del}|_{\Sigma}(X)$ , the surface discretisation error  $\epsilon(f)$  is a measure of the distance between the face  $f$  and the furthest adjacent point on the surface  $\Sigma$ . This measure can be thought of as a one-sided discrete Hausdorff distance,  $\epsilon(f) = H(\text{Del}|_{\Sigma}(X), \Sigma)$ , defined from a set of representative points on the triangulation  $\text{Del}|_{\Sigma}(X)$  to the surface  $\Sigma$ . Clearly, if the triangulation  $\text{Del}|_{\Sigma}(X)$  is a good piecewise representation of the surface  $\Sigma$ , the surface discretisation error  $\epsilon(f)$  should be small. See Figure 2 for a description of the surface discretisation error for a facet  $f \in \text{Del}|_{\Sigma}(X)$ .

**Definition 4.** Given a  $d$ -simplex  $\tau$ , its *radius-edge* ratio,  $\rho(\tau)$ , is given by:

$$\rho(\tau) = R/\|\mathbf{e}_0\|, \quad (1)$$

where  $R$  is the radius of the diametric ball of  $\tau$  and  $\|\mathbf{e}_0\|$  is the length of its shortest edge.

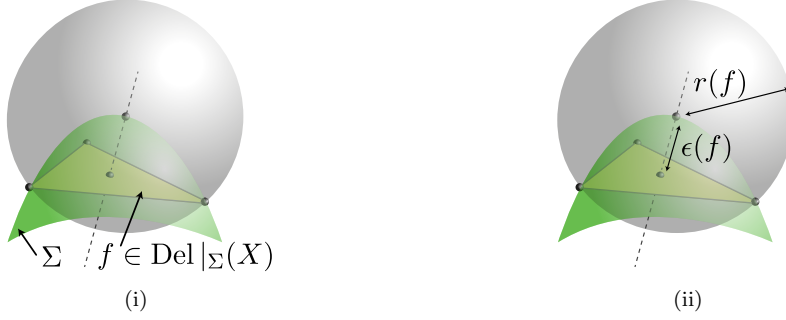
The radius-edge ratio is a measure of element shape-quality. It achieves a minimum,  $\rho(\tau) = 1/\sqrt{3}$ , for equilateral triangles and increases toward  $+\infty$  as elements tend toward degeneracy. For 2-simplexes, the radius-edge ratio is *robust* and can be related to the minimum plane angle  $\theta_{\min}$  between adjacent edges, such that  $\rho(\tau) = \frac{1}{2}(\sin(\theta_{\min}))^{-1}$ . Due to the summation of angles in a triangle, given a minimum angle  $\theta_{\min}$  the largest angle  $\theta_{\max}$  is clearly bounded, such that  $\theta_{\max} \leq 180^\circ - 2\theta_{\min}$ .

### 2.1. An Existing Delaunay-refinement Algorithm

The development of *provably-good* Delaunay-refinement schemes for surface-based mesh generation is an ongoing area of research. An algorithm for the meshing of closed 2-manifolds embedded in  $\mathbb{R}^3$ , adapted largely from the methods presented by Boissonnat and Oudot in [15, 16] is presented here. This method is largely equivalent to the CGALMESH algorithm, available as part of the CGAL package, and summarised by Jamin, Alliez, Yvinec and Boissonnat in [17]. A similar algorithm is also described by Cheng, Dey and Shewchuk in [19] and Dey and Levine in [24], while a variant developed for the reconstruction of medical images is presented by Foteinos, Chernikov and Chrisochoides in [25]. The algorithm presented in this section is referred to as the ‘conventional’ Delaunay-refinement approach throughout, due to its direct use of circumcentre-based Steiner vertices.

As per Jamin et al. [17], the development of the conventional Delaunay-refinement algorithm is *geometry-agnostic*, being independent of the specific description used for the underlying surface  $\Sigma$ . It is required only that the geometry support a so-called *oracle* predicate that can be used to compute the intersection of a given line segment with the surface  $\Sigma$ . The Frontal-Delaunay algorithm presented in Section 3, additionally requires that the oracle compute intersections between a disk of a given radius and the surface  $\Sigma$ . Such constructions will be discussed in further detail in Section 3. While a very general class of surface description

Figure 2: The surface Delaunay ball for a restricted 2-face  $f \in \text{Del}|_{\Sigma}(X)$ , showing (i) placement of the surface ball at the intersection of the dual edge  $\mathbf{v}_f \in \text{Vor}(X)$  and the surface  $\Sigma$ , and (ii) associated radius  $r(f)$  and surface discretisation error  $\epsilon(f)$ .



is supported at the theoretical level, in this study, attention is restricted to the development of so-called *remeshing* operations, in which the underlying surfaces  $\Sigma$  are specified as existing 2-manifold triangular complexes  $\mathcal{P}$ . This restriction is made in the present study for convenience only, and future work is intended to focus on more general surface descriptions, including implicit and analytic functions, in addition to those that contain sharp creases.

Following Jamin et al. [17], the Delaunay-refinement algorithm takes as input a surface domain, described by a smooth 2-manifold  $\Sigma$ , an upper bound on the allowable element radius-edge ratio  $\bar{\rho}$ , a mesh size function  $\bar{h}(\mathbf{x})$  defined at all points on the surface  $\Sigma$  and an upper bound on the allowable surface discretisation error  $\bar{\epsilon}(\mathbf{x})$ . The input surface  $\Sigma$  encloses a bounded volume  $\Omega$ . The algorithm returns a triangulation  $\mathcal{T}|_{\Sigma}$  of the surface  $\Sigma$ , where  $\mathcal{T}|_{\Sigma}$  is a restricted Delaunay surface triangulation of a point-wise sampling  $X \in \Sigma$ , such that  $\mathcal{T}|_{\Sigma} = \text{Del}|_{\Sigma}(X)$ . As a by-product, the algorithm also returns a coarse triangulation  $\mathcal{T}|_{\Omega}$  of the enclosed volume  $\Omega$ , where  $\mathcal{T}|_{\Omega}$  is a restricted Delaunay volume triangulation  $\mathcal{T}|_{\Omega} = \text{Del}|_{\Omega}(X)$ . Both  $\text{Del}|_{\Sigma}(X)$  and  $\text{Del}|_{\Omega}(X)$  are sub-complexes of the full-dimensional Delaunay tessellation  $\text{Del}(X)$ . Note that  $\text{Del}|_{\Sigma}(X)$  is a triangular complex, while  $\text{Del}|_{\Omega}(X)$  and  $\text{Del}(X)$  are tetrahedral complexes. The method is summarised in Algorithm 2.1.

The Delaunay-refinement algorithm guarantees, firstly, that all elements in the output surface triangulation  $f \in \mathcal{T}|_{\Sigma}$  satisfy element shape constraints,  $\rho(f) \leq \bar{\rho}$ , element size constraints  $h(f) \leq \bar{h}(\mathbf{x}_f)$  and surface discretisation bounds  $\epsilon(f) \leq \bar{\epsilon}(\mathbf{x}_f)$ . Furthermore, for sufficiently small mesh size functions  $\bar{h}(\mathbf{x})$ , the surface triangulation  $\mathcal{T}|_{\Sigma}$  is guaranteed to be a *good* piecewise approximation of the underlying surface  $\Sigma$ , exhibiting both geometrical and topological convergence as  $\bar{h}(\mathbf{x}) \rightarrow 0$ , consistent with the characteristics of restricted Delaunay tessellations outlined in Definition 1.

The Delaunay-refinement algorithm begins by creating an initial point-wise sampling of the surface  $X_0 \in \Sigma$ . Exploiting the discrete representation available for  $\Sigma$ , in this study the initial sampling is obtained as a *well-distributed* subset of the existing vertices  $Y \in \mathcal{P}$ , where  $\mathcal{P}$  is the poly-

hedral representation of the surface  $\Sigma$ . Care is taken to ensure that each connected component in  $\mathcal{P}$  is sampled. Alternative initialisation methods making use of robust *persistent-triangles* techniques are described in [16, 19].

In the next step, the initial triangulation objects are formed. In the present work, the full-dimensional Delaunay tessellation,  $\text{Del}(X)$ , is built using an incremental Delaunay triangulation algorithm, based on the Bowyer-Watson technique [26]. The restricted surface and volumetric triangulations,  $\text{Del}|_{\Sigma}(X)$  and  $\text{Del}|_{\Omega}(X)$ , are derived from  $\text{Del}(X)$  by explicitly testing for intersections between edges in the associated Voronoi diagram  $\text{Vor}(X)$  and the surface  $\Sigma$ . These queries are computed efficiently by storing the surface definition  $\mathcal{P}$  in a spatial-tree.

The main loop of the algorithm proceeds to incrementally refine any 2-faces  $f \in \text{Del}|_{\Sigma}(X)$  that violate either the radius-edge, element size or surface discretisation requirements. The refinement process is priority scheduled, with triangles  $f \in \text{Del}|_{\Sigma}(X)$  ordered according to their radius-edge ratios  $\rho(f)$ , ensuring that the element with the *worst* ratio is refined at each iteration. Individual elements are refined based on their surface Delaunay balls, with a triangle  $f \in \text{Del}(X)$  eliminated by inserting the centre of the largest ball  $B(\mathbf{c}, r)_{\max} = \text{SDB}(f)$  into the tessellation  $\text{Del}(X)$ . This process is a direct generalisation of the circumcentre-based insertion method associated with Ruppert's algorithm for planar domains [9, 10].

As a consequence of changes to the full-dimensional tessellation following the insertion of a new Steiner vertex, corresponding updates to the restricted triangulations  $\text{Del}|_{\Sigma}(X)$  and  $\text{Del}|_{\Omega}(X)$  are instigated, ensuring that all tessellation objects remain valid throughout the refinement process. The Delaunay-refinement algorithm terminates when all 2-faces  $f \in \text{Del}|_{\Sigma}(X)$  satisfy all radius-edge, size and surface discretisation thresholds, such that<sup>1</sup>  $\rho(f) \leq \bar{\rho}$ ,  $h(f) \leq \alpha \bar{h}(\mathbf{x}_f)$  and  $\epsilon(f) \leq \bar{\epsilon}(\mathbf{x}_f)$ , respectively, where the element size  $h(f)$  is proportional to the radius of the associated surface ball  $B(\mathbf{c}, r)_{\max} = \text{SDB}(f)$ , such

<sup>1</sup>The coefficient  $\alpha = 4/3$ , ensuring that the mean element size does not, on average, undershoot the desired target size  $\bar{h}(\mathbf{x}_f)$ .

that<sup>2</sup>  $h(f) = \sqrt{3}r$ , and the target size  $\bar{h}(\mathbf{x}_f)$  is sampled at the centre of the surface ball  $\mathbf{x}_f = \mathbf{c}$ .

### 3. Restricted Frontal-Delaunay Methods

Frontal-Delaunay algorithms are a hybridisation of advancing-front and Delaunay-refinement techniques, in which a Delaunay triangulation is used to define the topology of a mesh while new Steiner vertices are inserted in a manner consistent with advancing-front methodologies. In practice, such techniques have been observed to produce very high-quality meshes, inheriting the smooth, semi-structured vertex placement of pure advancing-front methods and the optimal mesh topology of Delaunay-based approaches. While Frontal-Delaunay methods have previously been used by a range of authors in the context of planar, volumetric and parametric surface meshing, including, for example studies by Üngör and Erten [27], Rebay [28], Mavriplis [29], Frey, Borouchaki and George [30], and Remacle, Henrotte, Carrier-Baudouin, Béchet, Marchandise, Geuzaine and Mouton [31], the authors are not aware of any previous investigations describing the application of such techniques to the surface meshing problem directly. The conventional advancing-front method, on the other hand, has been generalised to support surface meshing, as, for example, outlined in studies by Rypl [32, 6] and Schreiner, Scheidegger, Fleishman and Silva [7, 33].

In these previous studies, the conventional planar advancing-front methodology is extended directly to facilitate surface operations. Meshing proceeds via the incremental introduction of a well-distributed set of vertices  $X$  positioned on the surface  $X \in \Sigma$ . It should be noted that, contrary to the restricted Delaunay techniques introduced in previous sections, advancing-front methods typically maintain a partial, and possibly non-Delaunay manifold triangular complex  $\mathcal{T}|_{\Sigma}$  only – they do not construct a full-dimensional tessellation  $\mathcal{T}|_{\Omega}$ . It should also be noted that, in addition to the usual limitations associated with advancing-front strategies, serious issues of robustness often afflict these techniques in practice due to the difficulties associated with the reliable evaluation of the requisite geometric intersection and overlap predicates for sets of non-planar elements. Additionally, for highly curved and/or poorly separated surface definitions it is known to be difficult to ensure the topological correctness of the output mesh  $\mathcal{T}|_{\Sigma}$ . Schreiner et al. [7, 33] introduce a number of heuristic techniques in an effort to overcome these difficulties.

#### 3.1. Off-centres

The new Frontal-Delaunay algorithm is based primarily on a generalisation of ideas introduced by Rebay, who,

in [28], developed a *planar* Frontal-Delaunay algorithm in which new vertices are positioned along edges of the associated Voronoi diagram. Rebay showed that new vertices can be positioned on  $\text{Vor}(X)$  according to an *a priori* mesh size function  $\bar{h}(\mathbf{x})$ , a strategy broadly consistent with conventional advancing-front techniques. While his algorithm maintains a Delaunay triangulation  $\mathcal{T} = \text{Del}(X)$  of the current vertex set  $X$ , it is still fundamentally an advancing-front scheme – bereft of guarantees on the element shape quality  $\rho(\tau)$ . Rebay reported that his scheme produced very high-quality output, typically outperforming conventional Delaunay-refinement in practice.

Üngör and Erten have approached the problem from the flip-side, introducing the notion of *generalised* Steiner vertices for planar Delaunay-refinement methods. Their ‘off-centre’ vertices are points lying along edges in the associated Voronoi diagram, as per Rebay. In [34], Üngör showed that when refining a triangle  $\tau$ , its off-centre should be positioned, if possible, such that the new triangle  $\sigma$  adjacent to the shortest edge in  $\tau$  satisfies the desired shape-quality target, such that  $\rho(\sigma) \leq \bar{\rho}$ . Importantly, Üngör demonstrated that such a strategy typically leads to an improvement in the performance of Delaunay-refinement in practice, reducing the size of the output  $|\mathcal{T}|$ . Üngör extended the guarantees derived for Ruppert’s Delaunay-refinement algorithm to his off-centre technique and showed that such bounds are typically improved upon in practice. Off-centre refinement is based on Ruppert’s Delaunay-refinement framework directly, involving modifications to the procedure used to refine low-quality triangles only.

The use of *generalised* Delaunay-refinement strategies has also been explored by Chernikov, Chrisochoides and Foteinos in [35, 36], in which Steiner points are positioned within a set of so-called *selection-balls* adjacent to element circumcentres. In [35], Chernikov and Chrisochoides show that a family of ‘provably-good’ generalised two- and three-dimensional Delaunay-refinement schemes exist, and can be realised via the specification of appropriate selection-ball radii parameters. It is unclear whether this selection-ball formalism incorporates the full set of Voronoi-type off-centres currently in use, including those introduced by Rebay [28], and Erten and Üngör [27], in addition to the strategies defined in the present study. Such a determination may form the basis for future work.

#### 3.2. Point-placement Preliminaries

In this study, a generalisation of the ideas introduced by Rebay and Üngör is formulated for the surface meshing problem – using off-centre Steiner vertices to simulate the vertex placement strategy of a conventional advancing-front approach, while also preserving the framework of a restricted Delaunay-refinement technique. The aim of such a strategy is to recover the high element qualities and smooth, semi-structured point-placement generated by frontal methods, while inheriting the theoretical guarantees of Delaunay-refinement methods. Advancing-front algorithms typically incorporate a mesh size function  $\bar{h}(\mathbf{x})$ ,

<sup>2</sup>The coefficient  $\sqrt{3}$  represents the mapping between the edge length and diametric ball radius for an equilateral element. Such scaling ensures that size constraints are applied with respect to mean edge length.

---

**Algorithm 2.1** Surface mesh generation by restricted Delaunay-refinement

---

<pre> 1: <b>function</b> DELAUNAYSURFACE(<math>\Sigma, \Omega, \bar{\rho}, \bar{\epsilon}(\mathbf{x}), \bar{h}(\mathbf{x}), \mathcal{T} _{\Sigma}, \mathcal{T} _{\Omega}</math>) 2:   Form an initial sampling <math>X \in \Sigma</math> such that <math>X</math>    is <i>well-distributed</i> on <math>\Sigma</math>. 3:   Form Delaunay tessellation <math>\text{Del}(X)</math>. 4:   Form restricted objects <math>\text{Del} _{\Sigma}(X)</math> and <math>\text{Del} _{\Omega}(X)</math>. 5:   Enqueue all restricted 2-simplexes <math>Q _{\Sigma} \leftarrow</math> <math>f \in \text{Del} _{\Sigma}(X)</math>. A 2-simplex <math>f</math> is enqueued    if <math>\text{BADSIMPLEX}(f)</math> returns TRUE. 6:   <b>while</b> (<math>Q _{\Sigma} \neq \emptyset</math>) <b>do</b> <span style="float: right;"><math>\triangleright \{\text{main refinement sweeps}\}</math></span> 7:     Call <math>\text{REFINESIMPLEX}(f \leftarrow Q _{\Sigma})</math> 8:     Update the restricted Delaunay tessell-        ations <math>\text{Del} _{\Sigma}(X)</math> and <math>\text{Del} _{\Omega}(X)</math>. 9:     Update <math>Q _{\Sigma}</math> to reflect changes to <math>\text{Del} _{\Sigma}(X)</math>. 10:  <b>end while</b> 11:  <b>return</b> <math>\mathcal{T} _{\Sigma} = \text{Del} _{\Sigma}(X)</math> and <math>\mathcal{T} _{\Omega} = \text{Del} _{\Omega}(X)</math> 12: <b>end function</b> </pre>	<pre> 1: <b>function</b> REFINESIMPLEX(<math>f</math>) <span style="float: right;"><math>\triangleright \{\text{surface refinement}\}</math></span> 2:   Call <math>\text{SURFACEDELAUNAYBALL}(f, B(\mathbf{c}, r)_{\max})</math>. 3:   Form new Steiner vertex <math>\mathbf{p}</math> about <math>B(\mathbf{c}, r)_{\max}</math>. 4:   Insert Steiner vertex <math>X \leftarrow \mathbf{p}</math>, update <math>\text{Del}(X) \leftarrow X</math>. 5: <b>end function</b>  1: <b>function</b> SURFACEDELAUNAYBALL(<math>f</math>) <span style="float: right;"><math>\triangleright \{\text{surface ball}\}</math></span> 2:   Form Voronoi edge <math>\mathbf{v}_f</math> orthogonal to 2-simplex <math>f</math>. 3:   Form the set of associated surface Delaunay      balls <math>B(\mathbf{c}, r)_i</math> for the restricted 2-simplex <math>f \in</math> <math>\text{Del} _{\Sigma}(X)</math>. Balls are centred about the set of      surface intersections <math>\mathbf{v}_f \cap \Sigma \neq \emptyset</math>. 4:   <b>return</b> <math>B(\mathbf{c}, r)_{\max}</math>, where <math>r_{\max}</math> is maximal. 5: <b>end function</b>  1: <b>function</b> BADSIMPLEX(<math>f</math>) <span style="float: right;"><math>\triangleright \{\text{termination criteria}\}</math></span> 2:   <b>return</b> <math>\rho(f) &gt; \bar{\rho}</math> or <math>\epsilon(f) &gt; \bar{\epsilon}(\mathbf{x}_f)</math> or <math>h(f) &gt; \bar{h}(\mathbf{x}_f)</math> 3: <b>end function</b> </pre>
--	---

---

a function  $f : \mathbb{R}^3 \rightarrow \mathbb{R}^+$  defined over the domain to be meshed, where  $\bar{h}(\mathbf{x})$  represents the desired edge length  $\|\mathbf{e}\|$  at any point  $\mathbf{x} \in \Sigma$ . This mesh size function typically incorporates size constraints dictated by both the user and the geometry of the domain to be meshed. The construction of appropriate mesh size functions will be discussed in subsequent sections, but for now, it suffices to note that  $\bar{h}(\mathbf{x})$  is a  $g$ -Lipschitz function defined at all points on  $\Sigma$  with  $0 < g < 1$ .

The proposed Frontal-Delaunay algorithm is an extension of the restricted Delaunay-refinement algorithm presented in Section 2, modified to use off-centre rather than circumcentre-based refinement strategies. The basic framework is consistent with the Delaunay-refinement algorithm described previously, in which an initially coarse restricted Delaunay triangulation of a surface  $\Sigma$  is refined through the introduction of additional Steiner vertices  $X \in \Sigma$  until all constraints are satisfied. A restricted surface triangulation  $\mathcal{T}|_{\Sigma} = \text{Del}|_{\Sigma}(X)$  is constructed as a sub-complex of the full-dimensional tessellation  $\text{Del}(X)$  and a coarse restricted volumetric tessellation  $\mathcal{T}|_{\Omega} = \text{Del}|_{\Omega}(X)$  is also available as a by-product. The constraints satisfied by the Frontal-Delaunay algorithm are identical to those described previously for the restricted Delaunay-refinement scheme, with upper bounds on the radius-edge ratio  $\bar{\rho}$ , surface discretisation error  $\bar{\epsilon}(\mathbf{x}_f)$  and element size  $\bar{h}(\mathbf{x}_f)$  all required to be satisfied for convergence. See Algorithm 2.1 for a detailed outline of the method.

### 3.3. Point-placement Strategy

Given a surface facet  $f \in \text{Del}|_{\Sigma}(X)$  marked for refinement, the new Steiner vertex introduced to eliminate  $f$  is an off-centre, constructed to adhere to local size and shape constraints. The off-centres introduced in this study involve the placement of two distinct kinds of Steiner vertices. Type I vertices,  $\mathbf{c}^{(1)}$ , are equivalent to conventional element circumcentres, and are used to satisfy constraints on the element radius-edge ratios. Type II vertices,  $\mathbf{c}^{(2)}$ , are *size-optimal* points, designed to satisfy element sizing constraints in a locally optimal fashion. Adopting the

*generalised* off-centre framework introduced by Üngör, the ‘ideal’ location of the size-optimal off-centre  $\mathbf{c}^{(2)}$  is based on a consideration of the isosceles triangle  $\sigma$  formed about the short ‘frontal’ edge  $\mathbf{e}_0 \in f$ . The point  $\mathbf{c}^{(2)}$  is positioned to ensure that  $\sigma$  satisfies local size constraints.

Given a *refinable* 2-simplex  $f \in \text{Del}|_{\Sigma}(X)$ , the size-optimal Type II vertex  $\mathbf{c}^{(2)}$  is positioned at an intersection of the surface  $\Sigma$  and a plane  $\mathcal{V}$ , where  $\mathcal{V}$  is aligned with the local face of the Voronoi diagram  $\text{Vor}(X)$  associated with the frontal edge  $\mathbf{e}_0 \in f$ . The plane is positioned such that it passes through three local points on  $\text{Vor}(X)$ : the midpoint of the frontal edge  $\mathbf{e}_0 \in f$ , the centre of the diametric ball of  $f$  and the centre of the surface Delaunay ball  $B(\mathbf{c}, r)_{\max} = \text{SDB}(f)$ . The vertex  $\mathbf{c}^{(2)}$  is positioned such that it forms an isosceles triangle candidate  $\sigma$  about the frontal edge  $\mathbf{e}_0$ , such that its size  $h(\sigma)$  satisfies local constraints. Specifically, the altitude of  $\sigma$  is computed from local mesh-size information, such that

$$a_{\sigma} = \min((\bar{h}_{\sigma}^2 - \|\frac{1}{2}\mathbf{e}_0\|^2)^{\frac{1}{2}}, \sqrt{3}/2 \bar{h}_{\sigma}) \quad (2)$$

$$\bar{h}_{\sigma} = \frac{1}{2}(\bar{h}(\mathbf{m}_1) + \bar{h}(\mathbf{m}_2)) \quad (3)$$

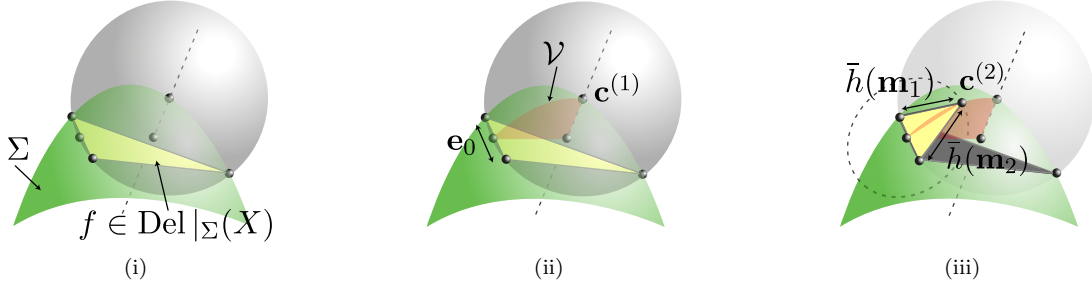
where the  $\mathbf{m}_i$ ’s are the edge midpoints and  $\sqrt{3}/2 \bar{h}_{\sigma}$  is the altitude for an ‘ideal’ element.

The position of the point  $\mathbf{c}^{(2)}$  is calculated by computing the intersection of the surface  $\Sigma$  with a circle of radius  $a_{\sigma}$ , centred at the midpoint of the frontal edge  $\mathbf{e}_0 \in f$  and inscribed on the plane  $\mathcal{V}$ . In the case of multiple intersections, the candidate point  $\mathbf{c}_i^{(2)}$  of closest alignment to the *frontal* direction vector  $\mathbf{d}_f$  is selected. Specifically, the point  $\mathbf{c}_i^{(2)}$  that maximises the scalar product  $(\mathbf{c}_i^{(2)} - \mathbf{m}_0) \cdot \mathbf{d}_f$  is chosen, where  $\mathbf{m}_0$  is the midpoint of the short edge  $\mathbf{e}_0 \in f$  and the frontal direction vector  $\mathbf{d}_f$  is taken from the midpoint  $\mathbf{m}_0$  to the centre of the surface ball  $B(\mathbf{c}, r)_{\max} = \text{SDB}(f)$ .

For non-uniform  $\bar{h}(\mathbf{x})$ , expressions for the position of the point  $\mathbf{c}^{(2)}$  are weakly non-linear, with the altitude  $a_{\sigma}$  depending on an evaluation of the mesh size function at the edge midpoints  $\bar{h}(\mathbf{m}_i)$  and visa-versa. In practice, since  $\bar{h}(\mathbf{x})$  is guaranteed to be Lipschitz smooth, a simple it-



Figure 3: Placement of off-centre vertices, showing (i) the surface ball associated with a facet  $f \in \text{Del}|\Sigma(X)$ , (ii) the plane  $\mathcal{V}$  aligned with the local facet of  $\text{Vor}(X)$  associated with the frontal edge  $\mathbf{e}_0 \in f$ , (iii) placement of the size-optimal vertex  $\mathbf{c}^{(2)}$ .



erative predictor-corrector procedure is sufficient to solve these expressions approximately. The positioning of size-optimal Type II Steiner vertices is illustrated in Figure 3.

Using the size-optimal Type II point  $\mathbf{c}^{(2)}$  and the Type I point  $\mathbf{c}^{(1)}$ , the final position of the refinement point  $\mathbf{c}$  for the facet  $f$  is calculated. The point  $\mathbf{c}$  is selected to satisfy the *limiting* local constraints, setting

$$\mathbf{c} = \begin{cases} \mathbf{c}^{(2)}, & \text{if } (d^{(2)} \leq d^{(1)}) \text{ and } (d^{(2)} \geq \frac{1}{2}\|\mathbf{e}_0\|), \\ \mathbf{c}^{(1)}, & \text{otherwise} \end{cases} \quad (4)$$

where  $d^{(1)} = \|\mathbf{c}^{(1)} - \mathbf{m}_0\|$  and  $d^{(2)} = \|\mathbf{c}^{(2)} - \mathbf{m}_0\|$  are distances from the midpoint of the frontal edge  $\mathbf{e}_0$  to the Type I and Type II vertices, respectively.

The selection criteria is designed to ensure that the refinement scheme smoothly degenerates to that of a conventional circumcentre-based Delaunay-refinement strategy in limiting cases, while using a locally shape-optimal approach where possible.

Specifically, the condition  $d^{(2)} \leq d^{(1)}$  guarantees that  $\mathbf{c}$  lies no further from the frontal edge  $\mathbf{e}_0$  than the centre of the surface Delaunay ball  $\text{SDB}(f)$ . Such behaviour ensures that a conventional circumcentre-based scheme is selected when the element size becomes sufficiently small. Additionally, the condition  $d^{(2)} \geq \frac{1}{2}\|\mathbf{e}_0\|$  ensures that the size-optimal scheme is only chosen when the edge  $\mathbf{e}_0$  is sufficiently small compared to the local mesh size function. This behaviour ensures that  $\mathbf{e}_0$  is a good ‘frontal’ edge candidate.

### 3.4. Refinement Order

In addition to the use of ‘off-centre’ point-placement schemes, the Frontal-Delaunay algorithm also introduces changes to the order in which elements are refined. To better mimic the behaviour of an advancing-front type method, elements are refined only if they are adjacent to an existing ‘frontal’ entity. In the case of surface facets  $f \in \text{Del}|\Sigma(X)$ , the short frontal edge  $\mathbf{e}_0 \in f$  must be shared by at least one adjacent facet  $f_j \in \text{Del}|\Sigma(X)$  that has ‘converged’ – satisfying its associated radius-edge, mesh-size and surface-error constraints. The idea of defining ‘frontal’ entities as a dynamic boundary between converged and

un-converged elements in the mesh is a common feature of Frontal-Delaunay algorithms, with similar approaches used in [28, 29, 30, 31].

### 3.5. Theoretical Guarantees

Boissonnat and Oudot [15, 16] have previously shown that the conventional restricted Delaunay-refinement algorithm presented in Section 2 is guaranteed to: (i) terminate in a finite number of steps, (ii) produce elements of bounded radius-edge ratios  $\rho(f) \leq \bar{\rho}$ , (iii) satisfy non-uniform user-defined mesh-sizing and surface error constraints  $h(f) \leq \bar{h}(\mathbf{x}_f)$ ,  $\epsilon(f) \leq \bar{\epsilon}(\mathbf{x}_f)$ , and (iv) provide a *good* geometrical and topological approximation to the underlying surface  $\Sigma$  when the mesh size function  $\bar{h}(\mathbf{x})$  is sufficiently small.

The behaviour of the new Frontal-Delaunay algorithm can be assessed using a similar framework.

### 3.6. Termination & Convergence

Firstly, the termination and convergence of the algorithm is analysed:

**Proposition 1.** *Given a smooth surface  $\Sigma$ , a radius-edge threshold  $\bar{\rho} \geq 1$  and positive mesh-size and surface-error functions  $\bar{h}(\mathbf{x}) \geq \bar{h}_0$ ,  $\bar{\epsilon}(\mathbf{x}) \geq \bar{\epsilon}_0$ ,  $\bar{h}_0, \bar{\epsilon}_0 \in \mathbb{R}^+$  the Frontal-Delaunay algorithm is guaranteed to terminate in a finite number of steps.*

Before seeking to prove Proposition 1 itself, a number of intermediate results are first established.

**Lemma 1.** *Let  $D \subset \mathbb{R}^d$  be a bounded domain. Given a subset  $X \subset D$ , satisfying  $\|\mathbf{u} - \mathbf{v}\| \geq \gamma$  for all pairs  $\mathbf{u}, \mathbf{v} \in X$  and some scalar  $\gamma \in \mathbb{R}^+$ , the size of  $X$  is bounded, such that  $|X| \leq \mu$  for some constant  $\mu \in \mathbb{Z}^+$ .*

This so-called *packing-lemma* states that any point-set  $X$  in a bounded domain  $\mathbb{R}^d$  for which there exists a positive minimum separation distance between the points in  $X$  must, consequently, be finite. As such, if it can be shown that a meshing algorithm preserves such a minimum separation length between its vertices, Lemma 1 can be invoked to prove that such a vertex-set is finite, and, as

a result, that termination of the algorithm is guaranteed. Lemma 1 is stated here without proof – interested readers are referred to [19] for full details.

Before examining the behaviour of the Frontal-Delaunay algorithm as a whole, the impact of a single refinement operation is analysed. Firstly, the behaviour of the various off-centre point-placement schemes introduced in Section 3.3 is examined:

**Proposition 2.** *Let the threshold on radius-edge ratios be  $\bar{\rho} \geq 1$  and  $\text{Del}|_{\Sigma}(X_k)$  be the restricted surface triangulation induced after  $k$  refinement operations. Given a low-quality surface facet  $f \in \text{Del}|_{\Sigma}(X_k)$ , for which  $\rho(f) \geq \bar{\rho}$ , the minimum edge length  $\|\mathbf{e}_0\| \in \text{Del}|_{\Sigma}(X_k)$  is not decreased following the insertion of a new Type I Steiner vertex at the centre of  $\text{SDB}(f)$ .*

*Proof.* Recalling that the surface facet  $f$  is locally Delaunay, the insertion of a new Type I vertex  $\mathbf{c}_k$  at the centre of the associated surface Delaunay ball  $B(\mathbf{c}_k, r_k) = \text{SDB}(f)$  is guaranteed to create edges no shorter than the radius of the ball of  $f$ :

$$\|\mathbf{e}_0\|_{k+1} \geq r_k. \quad (5)$$

Here  $\|\mathbf{e}_0\|_{k+1}$  is the length of the shortest edge created by the insertion of the new vertex  $\mathbf{c}_k$ . Dividing by the current minimum edge length, and using the definition of the radius-edge ratio:

$$\frac{\|\mathbf{e}_0\|_{k+1}}{\|\mathbf{e}_0\|_k} \geq \frac{r_k}{\|\mathbf{e}_0\|_k} \geq \bar{\rho}. \quad (6)$$

Clearly, when  $\bar{\rho} \geq 1$  the existing minimum length  $\|\mathbf{e}_0\|_k$  is preserved by the insertion of the new vertex  $\mathbf{c}_k$ .  $\square$

**Proposition 3.** *Let  $\bar{h}(\mathbf{x}) \geq \bar{h}_0$ ,  $\bar{h}_0 \in \mathbb{R}^+$  be a positive mesh-size function and  $\text{Del}|_{\Sigma}(X_k)$  be the restricted surface triangulation induced after  $k$  refinement operations. Given a refinable surface facet  $f \in \text{Del}|_{\Sigma}(X_k)$  use of the Type II point-placement scheme is ‘declined’ when  $f$  is sufficiently small. Specifically, selection of the Type II point-placement scheme requires that  $r_k \geq H$ , where  $r_k$  is the radius of the surface ball associated with the facet  $f$  and  $H$  is the length of the new frontal edge candidates defined via expressions (2)–(3).*

*Proof.* Given a surface facet  $f$ , the off-centre selection criteria expressed in (4) requires that the Type II off-centre  $\mathbf{c}^{(2)}$  lies along a ‘safe’ region of the Voronoi diagram – bounded by the centre of the ball  $B(\mathbf{c}_k, r_k) = \text{SDB}(f)$  such that:

$$\|\mathbf{c}^{(2)} - \mathbf{m}_0\| \leq \|\mathbf{c}_k - \mathbf{m}_0\| \quad (7)$$

where  $\mathbf{m}_0$  is the midpoint of the base edge  $\|\mathbf{e}_0\|$ . Expressing (7) in terms of the target edge lengths and surface ball radii:

$$\sqrt{H^2 - \|\frac{1}{2}\mathbf{e}_0\|^2} \leq \sqrt{r_k^2 - \|\frac{1}{2}\mathbf{e}_0\|^2} \quad (8)$$

where  $H$  is the desired edge-length, computed from local mesh-size constraints (2)–(3). This expressions can be further simplified, leading to a simple inequality for the selection of Type II vertices:

$$H \leq r_k \quad (9)$$

It can be seen that the Type II point-placement scheme is *declined* when the surface ball associated with a given facet  $f$  is sufficiently small compared to the local target edge-length  $H$ .  $\square$

Using Propositions 2 and 3, the off-centre point-placement strategies utilised in the Frontal-Delaunay algorithm are shown to constitute a ‘safe’ set of refinement operations, with the insertion of Type I vertices leading to a non-decreasing minimum edge-length and use of the Type II scheme declined once the mesh is sufficiently well refined. Using these results, termination of the full Frontal-Delaunay algorithm can be re-visited:

**PROOF OF PROPOSITION 1.** The Frontal-Delaunay algorithm refines any surface facet  $f \in \text{Del}|_{\Sigma}(X)$  if: (i) it is of poor shape quality, such that  $\rho(f) \geq \bar{\rho}$ , (ii) it is too large, such that  $h(f) \geq \alpha\bar{h}(\mathbf{x}_f)$ , or (iii) it violates the local surface discretisation error threshold, such that  $\epsilon(f) \geq \bar{\epsilon}(\mathbf{x}_f)$ .

- (i) Using Proposition 3, it is known that the introduction of Type II off-centres is declined once the mesh is sufficiently well refined. Type I points alone can therefore be relied upon to satisfy element shape-constraints.

Given a radius-edge threshold  $\bar{\rho} \geq 1$ , Proposition 2 states that refinement does not lead to a reduction in minimum edge length. Shape-based refinement therefore preserves the existing minimal edge length  $\|\mathbf{e}_0\|_k$ , associated with either the initial sampling  $X_0 \in \Sigma$ , or the insertion of some previous vertex  $\mathbf{x}_k \in X$  due to size- or surface-error-driven refinement.

- (ii) Noting that  $\bar{h}(\mathbf{x})$  is positive, with  $\bar{h}(\mathbf{x}) \geq \bar{h}_0$  for some  $\bar{h}_0 \in \mathbb{R}^+$ , size-driven refinement is clearly *declined* once the local element size is sufficiently small. Recalling that  $h(f) = \sqrt{3}r$ , where  $r$  is the radius of the surface ball associated with a given facet  $f$ , the mesh-sizing constraints  $h(f) \leq \alpha\bar{h}(\mathbf{x}_f)$  are satisfied once all  $r \leq (\alpha/\sqrt{3})\bar{h}(\mathbf{x}_f)$ .
- (iii) Noting that  $\bar{\epsilon}(\mathbf{x})$  is positive, with  $\bar{\epsilon}(\mathbf{x}) \geq \bar{\epsilon}_0$  for some  $\bar{\epsilon}_0 \in \mathbb{R}^+$ , it is clear that surface-error driven refinement is *declined* once the local element size is sufficiently small. Recalling that  $\epsilon(f)$  is the orthogonal distance between a given facet  $f$  and the centre of its associated surface ball  $B(\mathbf{c}, r) = \text{SDB}(f)$ , it is clear that  $\epsilon(f) \leq r$ , as  $\text{SDB}(f)$  must circumscribe the facet. In the worst-case, surface-error driven refinement is declined once all  $r \leq \epsilon(\mathbf{x}_f)$ .



Using (i)–(iii), the progression of the Frontal-Delaunay algorithm can be examined. Starting with an initial vertex distribution  $X_0 \in \Sigma$ , both Type I and Type II point-placement schemes are used to satisfy mesh-size and surface-error constraints – with refinement continuing until all surface balls are sufficiently small, as per (ii)–(iii). Any remaining low-quality elements in violation of the radius-edge constraints are subsequently refined through the insertion of Type I vertices alone. This final phase is guaranteed to preserve minimum edge-length.

Using Lemma 1, it is clear that the Frontal-Delaunay algorithm is guaranteed to produce a point-wise sampling  $X \in \Sigma$  of finite size and must terminate in a finite number of steps as a consequence.

Finite termination leads directly to a number of useful auxiliary guarantees on both the nature and quality of the output tessellation. Adopting the conventional terminology, surface meshes generated using the Frontal-Delaunay algorithm can be considered to be ‘provably-good’:

**Corollary 1.** *Given a smooth surface  $\Sigma$ , a radius-edge threshold  $\bar{\rho} \geq 1$  and positive mesh-size and surface-error functions  $\bar{h}(\mathbf{x}) > 0$ ,  $\bar{\epsilon}(\mathbf{x}) > 0$ , all surface facets  $f \in \mathcal{T}|_\Sigma$  in the restricted surface tessellation  $\mathcal{T}|_\Sigma = \text{Del}|_\Sigma(X)$  generated by the Frontal-Delaunay algorithm satisfy: (i)  $\rho(f) < \bar{\rho}$ , (ii)  $h(f) < \alpha \bar{h}(\mathbf{x}_f)$ , and (iii)  $\epsilon(f) < \bar{\epsilon}(\mathbf{x}_f)$ .*

*Proof.* The Frontal-Delaunay maintains a queue of ‘bad’ elements, containing any surface facets  $f \in \text{Del}|_\Sigma(X)$  that are in violation of one or more local constraints. Specifically, a given 2-face  $f \in \text{Del}|_\Sigma(X)$  is enqueued if: (i)  $\rho(f) \geq \bar{\rho}$ , (ii)  $h(f) \geq \alpha \bar{h}(\mathbf{x}_f)$ , or (iii)  $\epsilon(f) \geq \bar{\epsilon}(\mathbf{x}_f)$ . Given that termination is assured, it is clear that the refinement queue must become empty after a finite number of steps and that all facets in the resulting triangulation  $\text{Del}|_\Sigma(X)$  satisfy the requisite radius-edge ratio, element size and surface error constraints as a consequence.  $\square$

In addition to proofs of termination and convergence, it is also necessary to ensure the point-set  $X \in \Sigma$  remains *well-distributed* throughout the refinement process. Such behaviour requires that no spurious short edges be introduced within the initial refinement phase as the algorithm seeks to satisfy mesh-size and surface-error constraints.

**Proposition 4.** *Let  $\text{Del}|_\Sigma(X_k)$  be the current restricted surface triangulation. Given a large refinable surface facet  $f \in \text{Del}|_\Sigma(X_k)$ , the minimum edge length  $\|\mathbf{e}_0\|_k$  is decreased by an  $O(1)$  factor of  $1/\sqrt{3}$  in the worst-case following the insertion of a new Type I or Type II Steiner vertex.*

*Proof.* Considering the insertion of new Type I and Type II vertices separately:

- (i) Recalling that the surface facet  $f$  is locally Delaunay, the insertion of a new Type I vertex  $\mathbf{c}_k$  at the centre of the associated surface Delaunay ball  $B(\mathbf{c}_k, r_k) =$

$\text{SDB}(f)$  is guaranteed to create edges no shorter than the radius of the ball of  $f$ :

$$\|\mathbf{e}_0\|_{k+1} \geq r_k. \quad (10)$$

Here  $\|\mathbf{e}_0\|_{k+1}$  is the length of the shortest edge created by the insertion of the new vertex  $\mathbf{c}_k$ . Dividing by the current minimum edge length, and using the definition of the radius-edge ratio:

$$\frac{\|\mathbf{e}_0\|_{k+1}}{\|\mathbf{e}_0\|_k} \geq \frac{r_k}{\|\mathbf{e}_0\|_k} = \rho(f). \quad (11)$$

Clearly, the minimum edge length is decreased when any element with  $\rho(f) < 1$  is refined. Noting that  $\rho(f)$  achieves a minimum of  $1/\sqrt{3}$  when  $f$  is equilateral, it can be seen that insertion of a Type I Steiner vertex leads to a reduction in the minimum edge length by a factor of  $1/\sqrt{3}$  in the worst case.

- (ii) A Type II vertex  $\mathbf{c}_k$  positioned on a facet  $\mathcal{V} \subset \text{Vor}(X)$  associated with the short edge  $e_0$ , is equidistant to the vertices  $\mathbf{x}_i, \mathbf{x}_j \in e_0$ . Since  $\mathcal{V}$  is a local facet of the Voronoi complex, no other vertex  $\mathbf{x}_m \in X$  lies closer to the point  $\mathbf{c}_k$ .

Recalling that the Type II refinement strategy requires the diametric ball of the edge  $e_0$  remain empty, a worst case reduction in  $\|\mathbf{e}_0\|$  occurs when  $\mathbf{c}_k$  is positioned at the intersection of  $\mathcal{V}$  and the diametric ball of  $e_0$ , forming a right triangle. Noting that  $\mathcal{V}$  intersects  $e_0$  at its midpoint, the minimum edge length is decreased by a factor of  $1/\sqrt{2}$  in such cases.

Overall, Type I point-placement is a limiting case, reducing minimum edge length by an  $O(1)$  factor of  $1/\sqrt{3}$  in the worst case.  $\square$

### 3.7. Quality of Approximation

While the preceding results address the behaviour and performance of the underlying Delaunay-refinement algorithm, the approximation quality of the resulting mesh is also a key consideration. Using the framework introduced by Boissonnat and Oudot [15, 16], a variety of geometrical and topological guarantees are sought. Firstly, a number of results from [15, 16] are summarised:

**Definition 5.** Given a bounded domain  $\Omega$  enclosed by a surface  $\Sigma$ , the *medial-axis* of  $\Omega$  is the topological closure of the set of centres of *empty* balls that touch the surface  $\Sigma$  at more than one point.

Noting that the medial-axis lies at the ‘centre’ of any geometrical features in the domain induced by  $\Sigma$ , it is clear that the distance to the medial axis, denoted  $d_M(\mathbf{x})$  throughout, is a measure of the local ‘thickness’ of the domain. Using such considerations, Boissonnat and Oudot introduce the notion of *loose  $\epsilon$ -samples* to describe point-wise samplings  $X \in \Sigma$  that induce restricted surface triangulations  $\text{Del}|_\Sigma(X)$  exhibiting favourable geometrical and topological guarantees.

**Definition 6.** Let  $X \in \Sigma$  be a point-wise sampling of a smooth surface  $\Sigma$ . Let  $\mathcal{E}_V(X) \subseteq \text{Vor}(X)$  be the set of edges in the associated Voronoi complex. The sampling  $X$  is called a *loose  $\epsilon$ -sample* of  $\Sigma$  if: (i) for all intersections  $\mathbf{x} \in \{\mathcal{E}_V(X) \cap \Sigma\}$  the associated  $\epsilon$ -balls,  $B(\mathbf{c}, r)$ , are non-empty, such that  $X \cap B(\mathbf{x}, \epsilon d_M(\mathbf{x})) \neq \emptyset$  for some  $\epsilon \in \mathbb{R}^+$ , and (ii) the restricted triangulation  $\text{Del}|_\Sigma(X)$  includes at least one vertex on all connected components of the surface  $\Sigma$ .

Boissonnat and Oudot additionally establish the following geometrical and topological guarantees for restricted Delaunay tessellations  $\text{Del}|_\Sigma(X)$  induced by such loose  $\epsilon$ -samples:

**Proposition 5.** *Let  $X \in \Sigma$  be a loose  $\epsilon$ -sampling of a smooth surface  $\Sigma$ , with  $\epsilon \leq 0.08$ . The associated restricted surface triangulation  $\text{Del}|_\Sigma(X)$  exhibits the following properties: (i) the triangulation  $\text{Del}|_\Sigma(X)$  is homeomorphic to the surface  $\Sigma$ , (ii) for any 2-face  $f \in \text{Del}|_\Sigma(X)$  the angle between the facet normal  $\hat{\mathbf{n}}_f$  and the surface normal  $\hat{\mathbf{n}}_\Sigma$ , sampled at the vertices  $\mathbf{x} \in f$ , is  $O(\epsilon)$ , and (iii) the one-sided Hausdorff distance  $H(\text{Del}|_\Sigma(X), \Sigma)$ , sampled at the centre of the surface Delaunay balls  $B(\mathbf{c}, r)_{\max} = \text{SDB}(f)$  is  $O(\epsilon^2)$ .*

The reader is referred to [15, 16] or [19] for additional details and proofs. Boissonnat and Oudot go on to show that restricted Delaunay refinement, consistent with the algorithm presented in Section 2, is a ‘provably-good’ technique, being guaranteed to produce a loose  $\epsilon$ -sample of a surface  $\Sigma$  in a finite number of steps, and to generate a high-quality surface tessellation  $\text{Del}|_\Sigma(X)$  as a result.

The Frontal-Delaunay algorithm presented in this study derives from the same Delaunay-refinement framework developed in [15, 16], and inherits much of its theoretical robustness as a result. Using the properties of loose  $\epsilon$ -samples, the surface triangulations generated using the Frontal-Delaunay algorithm are guaranteed to be a good geometrical and topological approximation to the underlying surface – provided the mesh size function is sufficiently small:

**Proposition 6.** *Given a smooth surface  $\Sigma$  and an  $\epsilon$ -conforming size function  $(\alpha/\sqrt{3})\tilde{h}(\mathbf{x}) \leq \epsilon d_M(\mathbf{x})$ , the point-wise sampling  $X \in \Sigma$  generated by the Frontal-Delaunay algorithm is a loose  $\epsilon$ -sample. When  $\epsilon \leq 0.08$  the associated triangulation  $\text{Del}|_\Sigma(X)$  is a tight topological and geometrical approximation of the surface  $\Sigma$ .*

*Proof.* The Frontal-Delaunay algorithm refines any large surface facets, ensuring that  $h(f) \leq \alpha \tilde{h}(\mathbf{x}_f)$  for all  $f \in \text{Del}|_\Sigma(X)$ . Rearranging the previous expression, and substituting for the given mesh size constraints leads to  $h(f) = \sqrt{3}r \leq \alpha(\sqrt{3}/\alpha)\epsilon d_M(\mathbf{x})$ , which simplifies to  $r \leq 0.08 d_M(\mathbf{x})$  when  $\epsilon \leq 0.08$ . Recalling Definition 6, it is clear that  $X$  is a loose  $\epsilon$ -sample of  $\Sigma$ , and, via Proposition 5,  $\text{Del}|_\Sigma(X)$  is a tight geometrical and topological approximation of  $\Sigma$  as a result.  $\square$

## 4. Mesh Size Functions

The construction of high-quality mesh-size functions is an important aspect of both the restricted Delaunay-refinement and Frontal-Delaunay algorithms presented in Sections 2 and 3. A good mesh size function  $\tilde{h}(\mathbf{x})$  incorporates sizing constraints imposed by both the user and the geometry of the domain to be meshed. These contributions can be considered via two separate size functions, where  $\tilde{h}_u(\mathbf{x})$  represents user-defined sizing information and  $\tilde{h}_g(\mathbf{x})$  encapsulates geometry driven constraints. In this study, it is required that both  $\tilde{h}_u(\mathbf{x})$  and  $\tilde{h}_g(\mathbf{x})$  be piecewise linear functions defined on a supporting tetrahedral complex  $\mathcal{S}$ . Construction of appropriate user-defined functions  $\tilde{h}_u(\mathbf{x})$  is highly problem dependent and detailed formulations are not considered here.

As per the discussions presented in Section 3.5, it is known that the geometric mesh size function  $\tilde{h}_g(\mathbf{x})$  must be sufficiently small relative to the *local-feature-size* of the domain, denoted  $\text{lfs}(\mathbf{x})$ , to ensure that the resulting surface triangulation  $\text{Del}|_\Sigma(X)$  is both topologically and geometrically correct. Specifically, to guarantee that the resulting tessellation is a *loose  $\epsilon$ -sample*, it is required that  $\tilde{h}_g(\mathbf{x}) \leq 0.08 \text{lfs}(\mathbf{x})$ , where  $\text{lfs}(\mathbf{x})$  is the distance to the *medial-axis* of the bounded domain  $\Omega$ . In practice, such constraints are typically found to be overly restrictive [19], allowing for a relaxation of the coefficient  $\epsilon$ . In this study,  $\epsilon = 1/2$  is used throughout. The local-feature-size can be expressed as the distance from any point  $\mathbf{x}_i \in \Sigma$  to the closest point on the *medial-axis* of the domain. Given that  $\Sigma$  is represented as an underlying triangulation  $\mathcal{P}$  in this work, a discrete approximation to  $\text{lfs}(\mathbf{x})$  can be calculated directly via the method of Amenta and Bern [21, 37, 38]. For the sake of brevity, we do not describe these methods in full here, but simply note that they provide an efficient means to estimate  $\text{lfs}(\mathbf{x})$  at the vertices of the supporting complex  $\mathcal{S}$ .

Given an estimate of  $\text{lfs}(\mathbf{x})$  on the boundary of  $\mathcal{S}$ , it is possible to exact a degree of user-defined control on the resulting size function via a Lipschitz smoothing process. Following the work of Persson [39], a *gradient-limited* function  $\tilde{h}(\mathbf{x})$  can be constructed by limiting the variation in  $\tilde{h}(\mathbf{x})$  over the elements of  $\mathcal{S}$ . In this study, a scalar smoothing parameter  $g \in \mathbb{R}^+$  is used to globally, and isotropically limit variation, such that

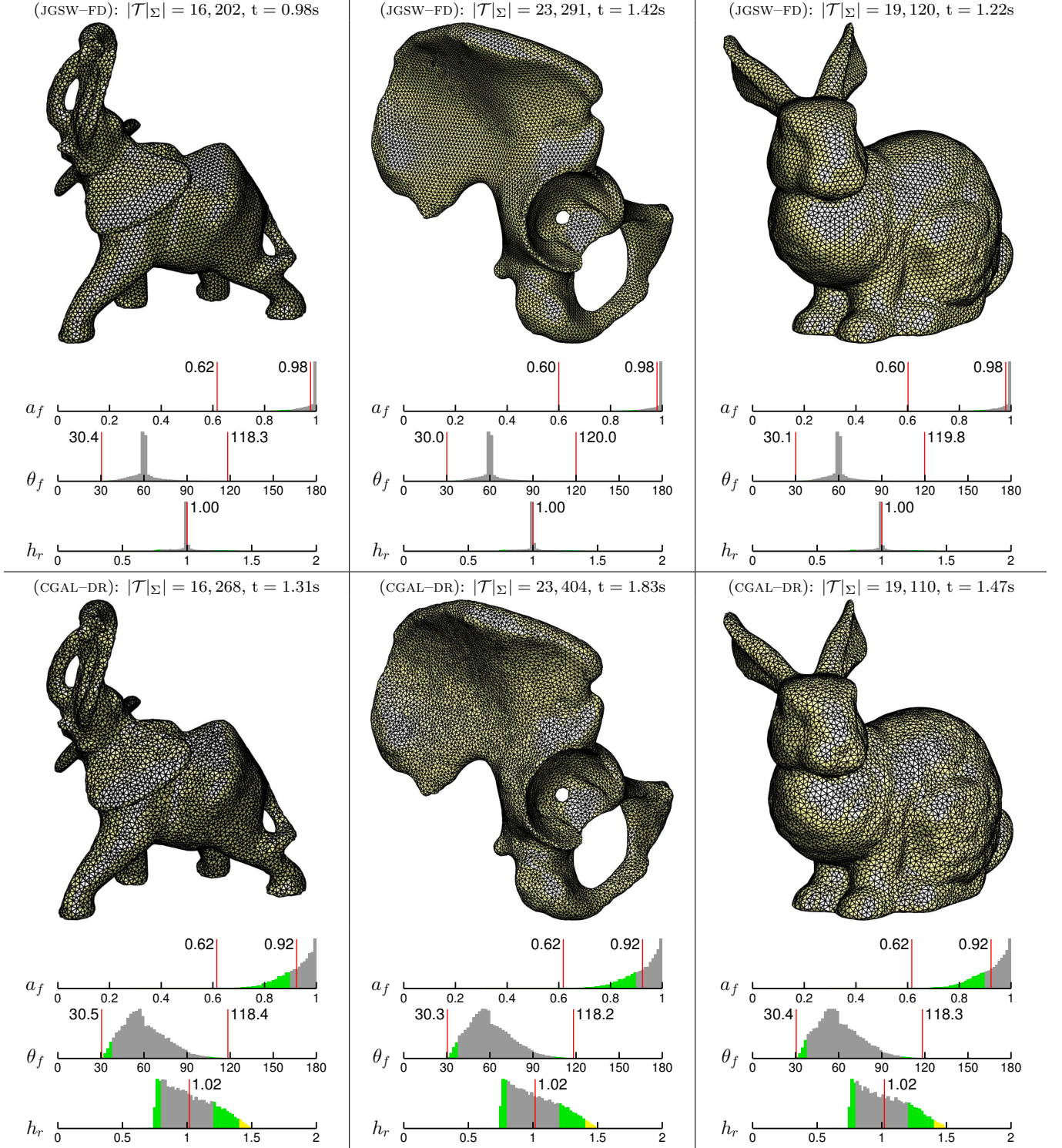
$$\tilde{h}(\mathbf{x}_i) \leq \tilde{h}(\mathbf{x}_j) + g \|\mathbf{x}_i - \mathbf{x}_j\| \quad (12)$$

for all vertex pairs  $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{S}$ . The gradient-limited size function  $\tilde{h}(\mathbf{x})$  becomes more uniform as  $g \rightarrow 0$ .

## 5. Results & Discussions

The performance of the Delaunay-refinement and Frontal-Delaunay surface meshing algorithms presented in Sections 2 and 3 was investigated experimentally, with both techniques used to mesh a series of benchmark problems.

Figure 4: Triangulations for the ELEPHANT, HIP and BUNNY problems, showing output for the JGSW-FD algorithm (upper) and the CGAL-DR algorithm (lower). Meshes were built using: (i) uniform mesh size functions  $\bar{h}(\mathbf{x}) = \alpha$ , with  $\alpha \in \mathbb{R}^+$ , (ii) tight constraints on element shape-quality, such that  $\theta_{\min} \geq 30^\circ$ , and (iii) uniform surface discretisation thresholds  $\bar{\epsilon} = (1/4) \bar{h}(\mathbf{x})$ . Element counts  $|\mathcal{T}|_\Sigma$ , and algorithm run-times (t) are included for each case. Normalised histograms of element area-length ratio  $a(f)$ , plane-angle  $\theta(f)$  and relative-length  $h_r$  are illustrated.



Both the Frontal-Delaunay and Delaunay-refinement algorithms were implemented, allowing the performance and output of the two techniques to be compared side-by-side. Due to similarities in the overall algorithmic structure, a common code-base was used, with the algorithms differing only in the type of Steiner vertices inserted, as per the discussions outlined in Section 3. Care was taken to ensure that both methods were implemented in a consistent fashion, allowing unbiased comparisons to be made between the algorithms without needing to account for systemic differences arising from particular implementation and/or design choices. Both algorithms were implemented in C++ and compiled as 64-bit executables. Both the Frontal-Delaunay and Delaunay-refinement algorithms have been implemented as part of the JIGSAW meshing package, currently available by request from the first author. These implementations are referred to as JGSW-FD and JGSW-DR throughout the following discussions, with the suffixes -FD and -DR denoting ‘Frontal-Delaunay’ and ‘Delaunay-refinement’, respectively.

In order to provide additional performance information, the well-known CGALMESH implementation [17] was also included in a subset of the meshing comparisons. The CGALMESH algorithm was sourced from version 4.6 of the CGAL package [40, 41] and was compiled as a 64-bit library. The CGALMESH algorithm is referred to as CGAL-DR throughout the following discussions, with the suffix -DR denoting ‘Delaunay-refinement’. All tests were run using a single core of an Intel i7 processor. Visualisation and post-processing was completed using MATLAB.

### 5.1. Preliminaries

The various Delaunay-refinement and Frontal-Delaunay algorithms were used to mesh a series of eight benchmark problems, including (i) a set of three test-cases presented in Figure 4, in which the performance of the new JGSW-FD algorithm and the existing CGAL-DR implementation was compared, (ii) three additional test-cases shown in Figure 5, designed to contrast the performance of the JGSW-FD and JGSW-DR algorithms, and (iii) two detailed comparative studies presented in Figures 6 and 7 designed to assess the impact of non-uniform mesh-sizing constraints on the performance of the JGSW-FD and JGSW-DR algorithms.

In all test cases, a constant radius-edge ratio threshold,  $\bar{\rho} = 1$  was specified, corresponding to  $\theta_{\min} \geq 30^\circ$ . Additionally, non-uniform surface discretisation constraints were enforced, setting  $\bar{\epsilon}(\mathbf{x}) = \beta \bar{h}(\mathbf{x})$ , with  $\beta = 1/4$ . For all test problems, detailed statistics on element quality are presented, including histograms of element *area-length*  $a(f)$ , *plane-angle*  $\theta(f)$  and *relative-length*  $h_r$  metrics. Histograms further highlight the minimum and mean area-length metrics, the worst-case plane angle bounds,  $\theta_{\min}$  and  $\theta_{\max}$  and the mean relative edge-length.

Overall, both Delaunay-refinement algorithms (CGAL-DR, JGSW-DR) and the new Frontal-Delaunay technique (JGSW-FD) were found to successfully mesh the full set

of benchmark problems, satisfying all constraints. These results demonstrate that the new Frontal-Delaunay technique is capable of generating high-quality surface triangulations, satisfying constraints on element shape-quality  $\rho(f)$ , element size  $h(\mathbf{x}_f)$  and surface discretisation error  $\epsilon(\mathbf{x}_f)$ , consistent with conventional Delaunay-refinement approaches.

### 5.2. Mesh Quality Metrics

Before moving on to a detailed analysis of results, a number of mesh quality metrics are first introduced.

**Definition 7.** Given a 2-simplex  $f$ , the *plane-angle* between any two adjacent edges is given by:

$$\cos(\theta_{i,j}) = \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\| \|\mathbf{e}_j\|} \quad (13)$$

for all adjacent pairs  $\mathbf{e}_i, \mathbf{e}_j \in f$ .

Clearly, high-quality surface triangulations include a majority of element plane-angles clustered about  $60^\circ$ .

**Definition 8.** Given a 2-simplex  $f$ , its *area-length* ratio,  $a(f)$ , is given by:

$$a(f) = \frac{4\sqrt{3}}{3} \frac{A_f}{\|\mathbf{e}_{\text{rms}}\|^2}, \quad (14)$$

where  $A_f$  is the signed-area of  $f$  and  $\|\mathbf{e}_{\text{rms}}\|$  is the root-mean-square edge length.

The area-length ratio is a robust, scalar measure of element shape-quality, and is typically normalised to achieve a score of +1 for *ideal* elements. The area-length ratio decreases with increasing distortion, achieving a score of +0 for degenerate elements and -1 for fully inverted elements.

**Definition 9.** Given a 2-simplex  $f$ , the *relative-length* of its edges is given by:

$$h_r(e_i) = \frac{\|\mathbf{e}_i\|}{\bar{h}(\mathbf{x}_e)} \quad (15)$$

where  $\|\mathbf{e}_i\|$  is the length of the  $i$ -th edge and  $\bar{h}(\mathbf{x}_e)$  is the mesh size function sampled at the edge midpoint.

The *relative-length* distribution  $h_r$  is a measure of size-function conformance, expressing the ratio of actual-to-desired edge length for all edges  $e \in \text{Del}|_{\Sigma}(X)$ . A value of  $h_r = 1$  indicates perfect conformance.

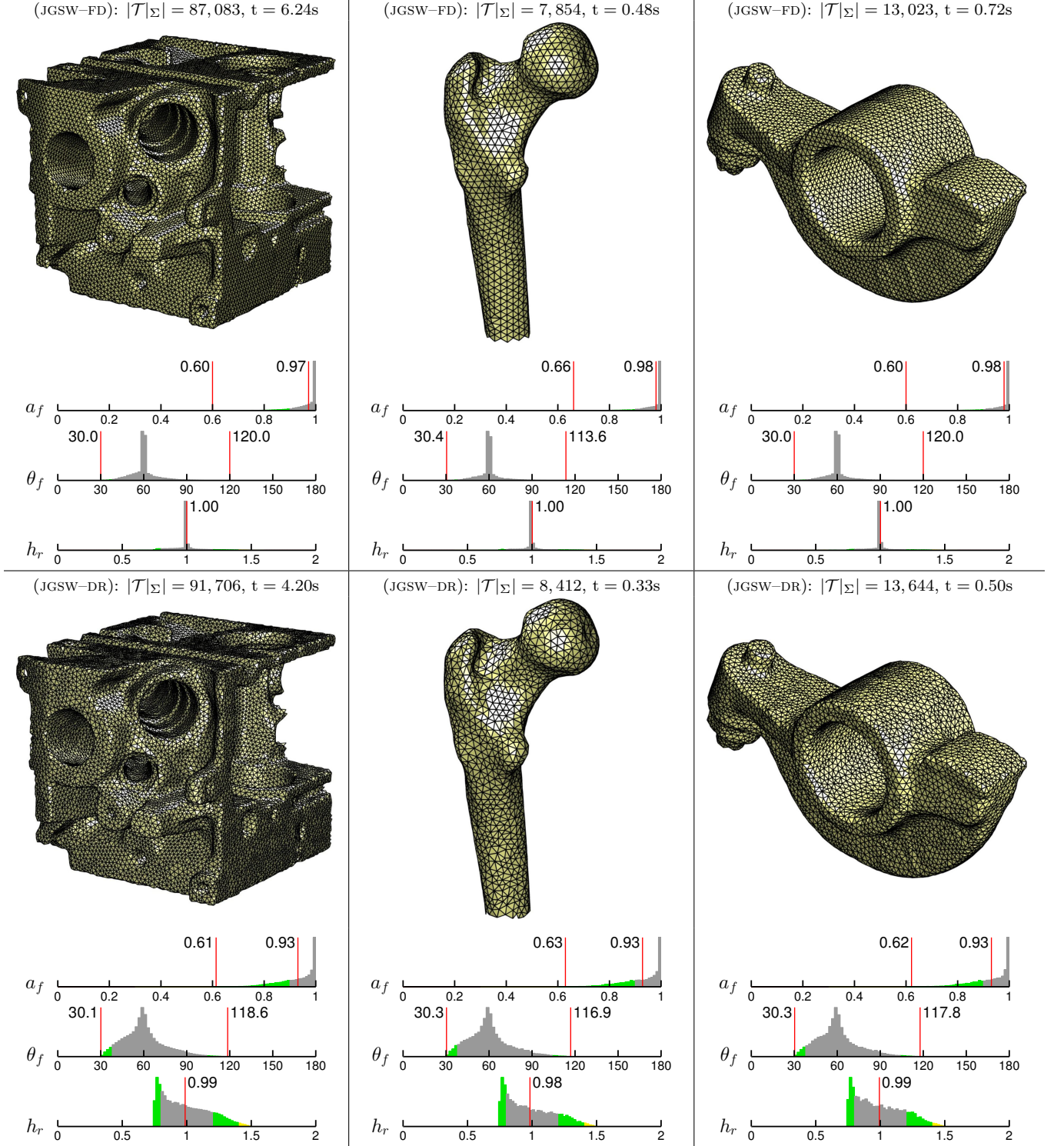
**Definition 10.** Given a distribution of element-wise plane angles  $\theta(f_i)$ , the Mean Absolute Deviation  $\text{MAD}(\theta_f)$  is given by:

$$\text{MAD}(\theta_f) = \frac{1}{n} \sum_{j=1}^n |\bar{\theta}(f_i) - \theta_j| \quad (16)$$

where  $\bar{\theta}(f_i)$  denotes the mean value of the underlying distribution  $\theta(f_i)$ .



Figure 5: Triangulations for the IFP2, FEMUR and ROCKER test problems, showing output for the JGSW-FD (upper) and JGSW-DR (lower) algorithms. Meshes were built using: (i) uniform mesh size functions  $\bar{h}(\mathbf{x}) = \alpha$ , with  $\alpha \in \mathbb{R}^+$ , (ii) tight constraints on element shape-quality, such that  $\theta_{\min} \geq 30^\circ$ , and (iii) non-uniform surface discretisation functions  $\bar{\epsilon} = (1/4)\bar{h}(\mathbf{x})$ . Element counts  $|\mathcal{T}|_\Sigma$ , and algorithm run-times (t) are included for each case. Normalised histograms of element area-length ratio  $a(f)$ , plane-angle  $\theta(f)$  and relative-length  $h_r$  are also illustrated.



$\text{MAD}(\theta_f)$  is a measure of the *spread* of the distribution of element angles, with smaller values indicative of distributions that are more tightly clustered about the mean value. In this study,  $\text{MAD}(\theta_f)$  is used as a measure of *average* element quality, with smaller values associated with higher quality meshes.

### 5.3. A Comparison of JGSW-FD and CGAL-DR

The performance of the JGSW-FD and CGAL-DR algorithms was assessed using a set of three test-cases. The ELEPHANT, HIP and BUNNY geometries were meshed using uniform mesh-size constraints, with a small constant value,  $\bar{h}(\mathbf{x}) = \alpha$ , imposed globally, where  $\alpha$  was chosen to be approximately 2% of the mean bounding-box dimension associated with each model. Due to differences in the way that mesh-size constraints are interpreted by the two meshing packages, the mesh-size value selected for the CGAL-DR algorithm was reduced by a factor of  $4/3$ . This reduction accounts for the fact that in CGAL-DR, mesh-size constraints are imposed with respect to the element circumradii, whereas the JGSW-FD algorithm treats the mesh-size function in terms of element edge length. Both algorithms were found to produce meshes incorporating consistent mean edge-lengths based on these modified mesh-size values.

Overall, the results shown in Figure 4 demonstrate that both the JGSW-FD and CGAL-DR algorithms generate high-quality surface meshes for all test cases – satisfying the required element shape-quality, mesh-size and surface discretisation error thresholds. Focusing on the distribution of element shape-quality explicitly, it is clear that the JGSW-FD algorithm achieves significantly better results, producing meshes with higher mean area-length ratios in all cases. In fact, with mean area-length ratios of  $\bar{a}(f) \simeq 0.98$  (compared to  $\bar{a}(f) \simeq 0.92$  for CGAL-DR), it is clear that the vast majority of elements generated by the JGSW-FD algorithm are of near-ideal shape. This improvement in mean mesh-quality is also evident in the distribution of element plane-angles, with the JGSW-FD approach generating histograms clustered more tightly about  $60^\circ$ . These results are confirmed by an analysis of the mean-absolute-deviation in angle,  $\text{MAD}(\theta_f)$  – included in Table 1, showing that the *spread* of the plane-angle distribution is reduced by a factor of approximately three through use of the JGSW-FD algorithm. Finally, analysis of the relative-length distributions shows that meshes generated using the JGSW-FD algorithm tightly conform to the imposed mesh size constraints, with a tight clustering of  $h_r$  about 1. In contrast, output generated using the CGAL-DR scheme is seen to incorporate significant sizing ‘error’, typified by broad distributions of relative-length, straddling  $h_r \simeq 1$ .

In terms of computational expense, meshes generated by the JGSW-FD and CGAL-DR algorithms are shown to be very similar in size. Additionally, despite the additional work required to support the off-centre refinement scheme,

the JGSW-FD algorithm appears to be slightly more efficient than CGAL-DR, though the difference in run-time performance is not significant.

### 5.4. A Comparison of JGSW-FD and JGSW-DR

Consistent with the analysis presented in Section 5.3, the performance of the JGSW-FD and JGSW-DR algorithms was next assessed. While CGAL-DR and JGSW-DR are representative of the same class of meshing *algorithms*, the benchmarks included in this section allow for an unbiased comparison of the Frontal-Delaunay and Delaunay-refinement approaches, with both the JGSW-FD and JGSW-DR algorithms benefiting from the same set of implementation design and optimisation decisions.

Consistent with the previous analysis, the performance of the JGSW-FD and JGSW-DR algorithms was assessed using a set of three test-cases. The IFP2, FEMUR and ROCKER geometries shown in Figure 5 were meshed using uniform mesh-size constraints, with a small constant value,  $\bar{h}(\mathbf{x}) = \alpha$ , imposed globally, where  $\alpha$  was chosen to be approximately 2% of the mean bounding-box dimension associated with each model. An analysis of the distributions of element area-length, plane-angle and relative edge-length measures confirms much the behaviour noted in Section 5.3 – in terms of overall mesh-quality and size-conformance, the JGSW-FD algorithm is a clear winner.

An analysis of computational expense leads to different conclusions, with the JGSW-DR algorithm seen to produce meshes that are slightly and yet consistently larger than those generated using JGSW-FD. The JGSW-DR scheme also appears to be somewhat faster than the JGSW-FD algorithm, typically requiring only approximately 70% of the run-time imposed by JGSW-FD. Such behaviour is not unexpected, with the additional work associated with the off-centre point-placement scheme employed in JGSW-FD expected to result in lower overall performance. The search for additional efficiency in the JGSW-FD algorithm is an interesting topic for future research.

### 5.5. Non-uniform Mesh-Size Constraints

A sequence of meshes for the BIMBA and KISS problems are presented in Figures 6 and 7, examining the impact of non-uniform mesh-size constraints on algorithm performance. A set of meshes were generated for both test cases using low, medium and high resolution settings, where the associated mesh size functions  $\bar{h}(\mathbf{x})$  were built using increasingly stringent gradient limits, such that  $g_i \in \{3/10, 2/10, 1/10\}$  respectively. Analysis of Figures 6 and 7 show that both the JGSW-FD and JGSW-DR algorithms generate high-quality meshes for all test cases – satisfying the required element quality, size and surface error thresholds. Analysis of the area-length and plane-angle distributions again shows that the JGSW-FD algorithm consistently outperforms the JGSW-DR scheme, generating meshes with higher mean area-length ratios in all cases. Furthermore, it is evident that the quality of meshes generated using the

Figure 6: Size-driven meshing for the BIMBA problem, showing output for the JGSW-FD (upper) and JGSW-DR (lower) algorithms. Meshes were built using: (i) graded,  $g$ -Lipschitz smooth mesh size functions  $\bar{h}(\mathbf{x})$ , with  $g_i \in \{3/10, 2/10, 1/10\}$  from left to right, (ii) tight constraints on element shape-quality, such that  $\theta_{\min} \geq 30^\circ$ , and (iii) non-uniform surface discretisation functions  $\bar{\epsilon} = (1/4) \bar{h}(\mathbf{x})$ . Element counts  $|\mathcal{T}|_\Sigma$ , and algorithm run-times (t) are included for each case. Normalised histograms of element area-length ratio  $a(f)$ , plane-angle  $\theta(f)$  and relative-length  $h_r$  are illustrated.

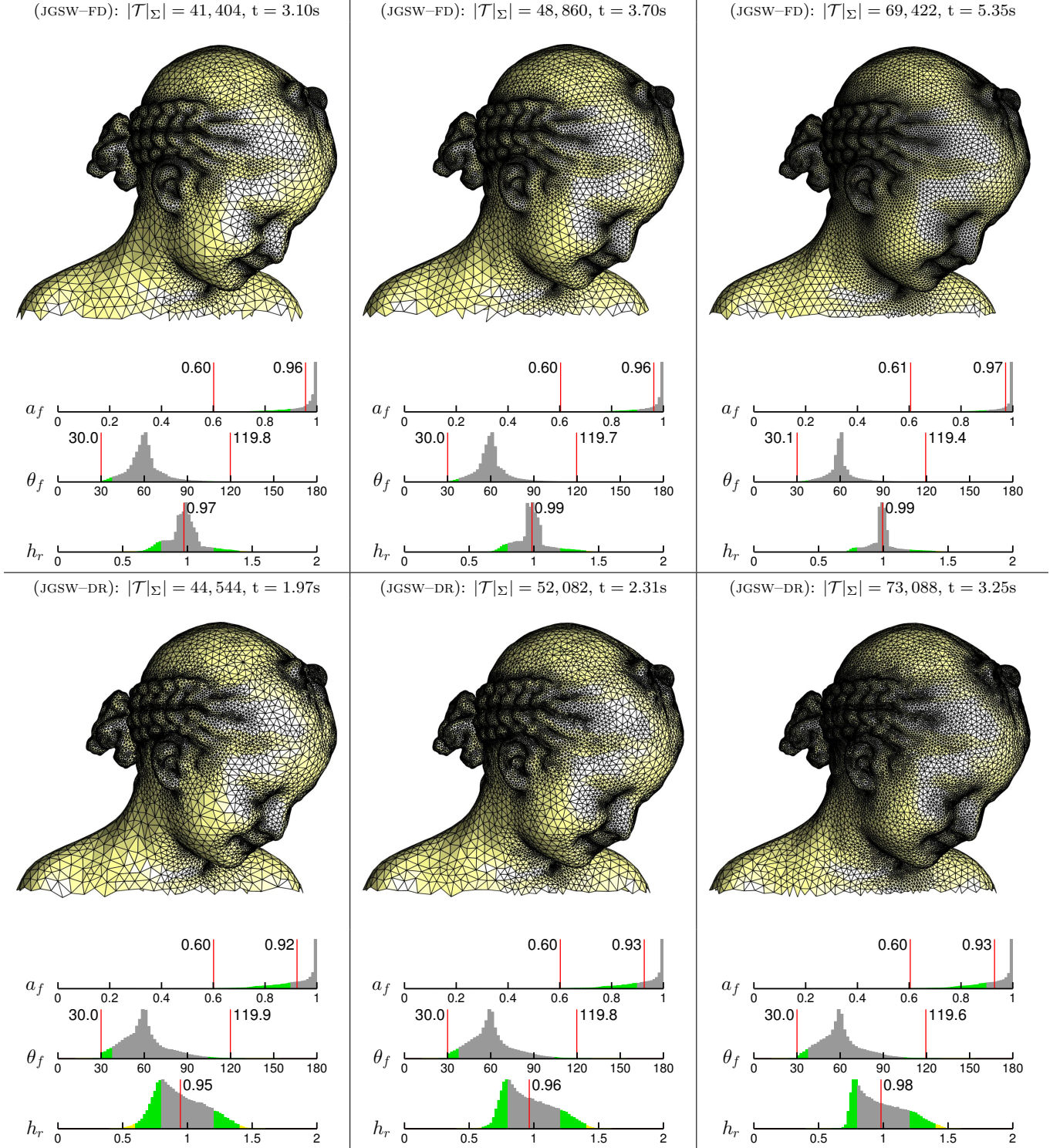




Figure 7: Size-driven meshing for the KISS problem, showing output for the JGSW-FD (upper) and JGSW-DR (lower) algorithms. Meshes were built using: (i) graded,  $g$ -Lipschitz smooth mesh size functions  $\bar{h}(\mathbf{x})$ , with  $g_i \in \{3/10, 2/10, 1/10\}$  from left to right, (ii) tight constraints on element shape-quality, such that  $\theta_{\min} \geq 30^\circ$ , and (iii) non-uniform surface discretisation functions  $\bar{\epsilon} = (1/4)\bar{h}(\mathbf{x})$ . Element counts  $|\mathcal{T}|_\Sigma$ , and algorithm run-times (t) are included for each case. Normalised histograms of element area-length ratio  $a(f)$ , plane-angle  $\theta(f)$  and relative-length  $h_r$  are illustrated.

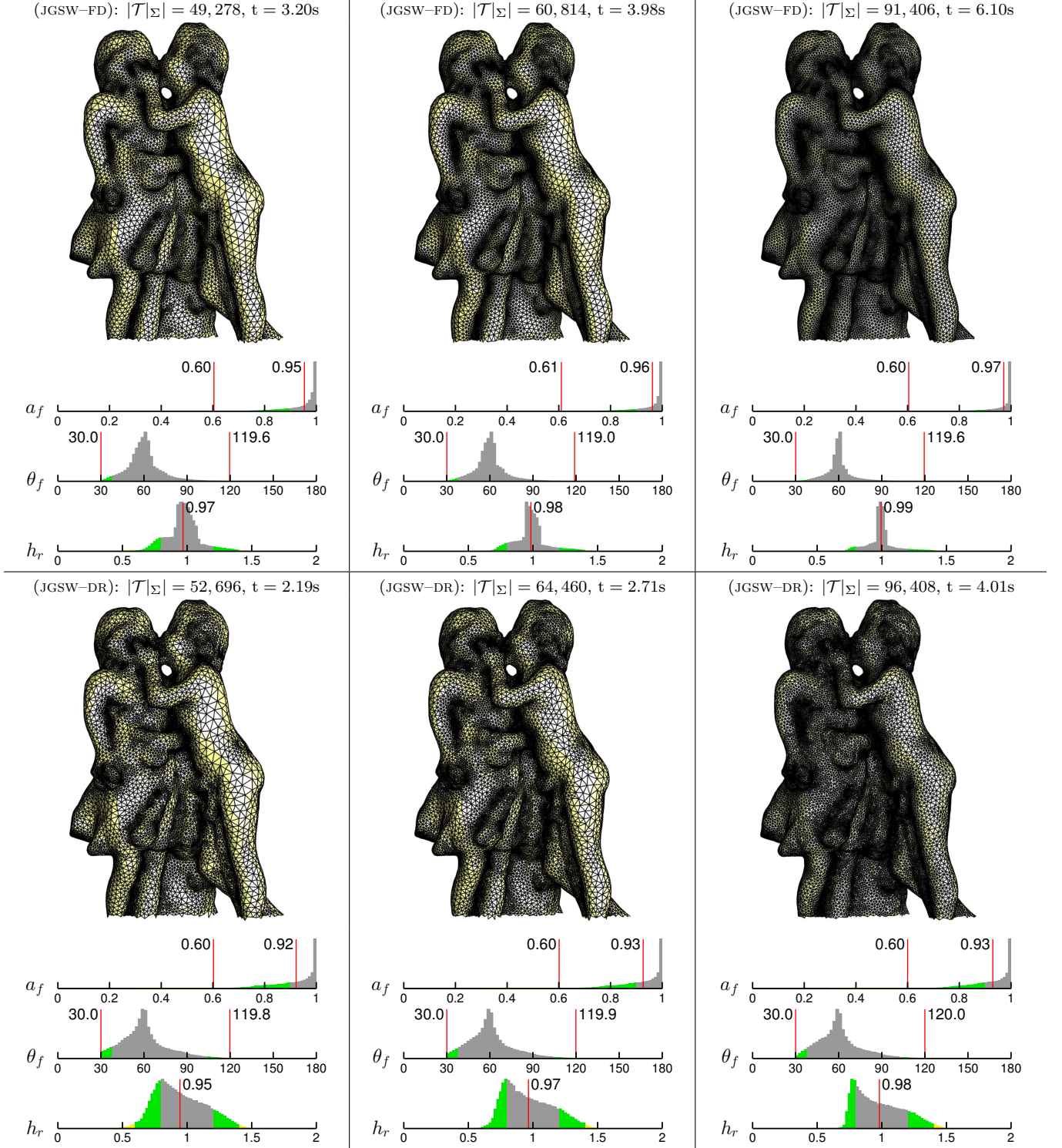


Table 1: A comparison of the mean absolute deviation in the element angle distributions  $\text{MAD}(\theta_f)$  for the various combinations of refinement algorithms and benchmark problems considered in this study. Smaller values correspond to improved mean mesh quality. CGAL-DR denotes the Delaunay-refinement algorithm included in the CGAL package, and JGSW-DR/JGSW-FD denote the Delaunay-refinement and Frontal-Delaunay algorithms implemented in the current study. The rightmost column indicates the reduction in absolute deviation achieved using the Frontal-Delaunay algorithm.

	$g$	CGAL-DR	JGSW-DR	JGSW-FD	FACTOR
ELEPHANT	–	11.96°	–	4.35°	2.8
HIP	–	11.90°	–	3.89°	3.1
BUNNY	–	12.02°	–	4.10°	2.9
IFP2	–	–	10.72°	5.14°	2.1
FEMUR	–	–	10.93°	3.85°	2.8
ROCKER	–	–	10.69°	3.92°	2.7
BIMBA	3/10	–	11.25°	7.83°	1.4
	2/10	–	10.99°	7.04°	1.6
	1/10	–	10.74°	5.68°	1.9
KISS	3/10	–	11.37°	8.15°	1.4
	2/10	–	10.98°	7.16°	1.5
	1/10	–	10.74°	5.74°	1.9

Frontal-Delaunay algorithm improves with increased uniformity, via  $g \rightarrow 0$ , as indicated by the increase in mean  $a(f)$  and the narrowing of the  $\theta(f)$  distribution about  $60.0^\circ$ . Results for the mean absolute deviation in the element angle distribution,  $\text{MAD}(\theta_f)$  – included in Table 1, shows that use of the JGSW-FD algorithm results in a reduction in the spread of the element angle distributions by a factor ranging from approximately 1.5 to 2.0.

In contrast, similar analysis shows that output generated via the JGSW-DR scheme to be essentially independent of sizing uniformity, with distributions of  $a(f)$  and  $\theta(f)$  showing little variation with  $\bar{h}(\mathbf{x})$ . Visually, the enhanced quality of the meshes generated using the Frontal-Delaunay algorithm is again evident, with all test cases showing a marked increase in both smoothness and sub-structure. Given the tight bounds on radius-edge ratios in these tests, the Delaunay-refinement algorithm appears to achieve a maximum mean area-length ratio  $\bar{a}(f) \simeq 0.92$ . The Frontal-Delaunay algorithm is seen to generate consistently better output, with mean area-length metrics reaching  $\bar{a}(f) \simeq 0.97$  at high resolution settings.

Consistent with previous results, analysis of the distribution of relative-length ratios show that the JGSW-FD algorithm tightly conforms to the imposed non-uniform sizing constraints, with  $h_r$  tightly clustered about 1. Furthermore, like the element shape-quality results discussed previously, it can be seen that this distribution improves as  $g \rightarrow 0$ , as indicated by the narrowing of the distribution about  $h_r = 1$ . In contrast, the Delaunay-refinement scheme is again shown to be consistently associated with significant sizing error, as illustrated by broad distributions of relative-length straddling  $h_r \simeq 1$ . Such behaviour appears to be largely independent of the characteristics of the imposed mesh size constraints.

## 6. Conclusions

A new Frontal-Delaunay algorithm has been developed to triangulate smooth surfaces embedded in  $\mathbb{R}^3$ . The new algorithm is based on the so-called *restricted* Delaunay paradigm, in which a surface mesh  $\text{Del}|_\Sigma(X)$ , conforming to an underlying surface description  $\Sigma$ , is constructed as a subset of a full-dimensional Delaunay tessellation  $\text{Del}(X)$ . The new restricted Frontal-Delaunay algorithm is based on a generalisation of the work of Rebay [28] and Üngör [34] for planar problems, in which generalised ‘off-centre’ Steiner vertices are inserted along edges in the Voronoi diagram. This work has been extended to support surface meshing operations through the development of a new point-placement strategy that positions vertices on facets in the underlying Voronoi complex. This new scheme allows for the insertion of both size- and shape-optimal Steiner vertices, leading to a hybrid approach that combines many of the advantages of conventional advancing-front and Delaunay-refinement techniques. A series of comparative experimental studies confirm the effectiveness of this new approach in practice, demonstrating that an improvement in mesh-quality is typically achieved when compared to conventional Delaunay-refinement schemes. Importantly, it has also been demonstrated that the new Frontal-Delaunay algorithm satisfies the same set of constraints as conventional restricted Delaunay-refinement approaches, adhering to limits on element radius-edge ratios, edge length and surface discretisation error. Results show that the new algorithm is an effective hybridisation of existing mesh generation techniques, combining the high element quality and mesh sub-structure of advancing-front techniques with the theoretical guarantees of Delaunay-refinement schemes. It is expected that applications that place a premium on mesh-quality, including problems in computational fluid dynamics and/or structural analysis, may benefit from the new Frontal-Delaunay technique. Fu-

ture work should focus on support for an extended class of surface definitions, including domains containing sharp features.

## Acknowledgements

This work was carried out at the University of Sydney with the support of an Australian Postgraduate Award. The authors also wish to thank the anonymous reviewers for their helpful comments and feedback.

## References

### References

- [1] D. Engwirda, D. Ivers, Face-centred Voronoi refinement for surface mesh generation, *Procedia Engineering* 82 (2014) 8–20.
- [2] M. Bern, D. Eppstein, J. Gilbert, Provably Good Mesh Generation, *J. Comput. Syst. Sci.* 48 (1990) 231–241.
- [3] S. A. Mitchell, S. A. Vavasis, Quality Mesh Generation in Three-dimensions., in: *Symposium on Computational Geometry*, 1992, pp. 212–221.  
URL <http://dblp.uni-trier.de/db/conf/compgeom/compgeom92.html#MitchellV92>
- [4] J. Peraire, J. Peiró, K. Morgan, Advancing-Front Grid Generation, in: J. F. Thompson, B. K. Soni, N. P. Weatherill (Eds.), *Handbook of Grid Generation*, Taylor & Francis, 1998.  
URL <http://books.google.com.au/books?id=ImaDT6ijKq4C>
- [5] J. Schöberl, NETGEN: An Advancing Front 2D/3D Mesh Generator based on Abstract Rules, *Computing and Visualization in Science* 1 (1) (1997) 41–52. doi:10.1007/s007910050004.  
URL <http://dx.doi.org/10.1007/s007910050004>
- [6] D. Rypl, Approaches to Discretization of 3D Surfaces, in: *CTU Reports*, Vol. 7, CTU Publishing House, Prague, Czech Republic, 2003.
- [7] J. Schreiner, C. E. Scheidegger, S. Fleishman, C. T. Silva, Direct (Re)Meshing for Efficient Surface Processing, *Computer Graphics Forum* 25 (3) (2006) 527–536. doi:10.1111/j.1467-8659.2006.00972.x.  
URL <http://dx.doi.org/10.1111/j.1467-8659.2006.00972.x>
- [8] L. P. Chew, Guaranteed-quality Triangular Meshes, Tech. rep., Cornell University, Department of Computer Science, Ithaca, New York (1989).
- [9] J. Ruppert, A New and Simple Algorithm for Quality 2-dimensional Mesh Generation, in: *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, SODA '93, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1993, pp. 83–92.  
URL <http://dl.acm.org/citation.cfm?id=313559.313615>
- [10] J. Ruppert, A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation, *Journal of Algorithms* 18 (3) (1995) 548–585. doi:http://dx.doi.org/10.1006/jagm.1995.1021.  
URL <http://www.sciencedirect.com/science/article/pii/S0196677485710218>
- [11] J. R. Shewchuk, Delaunay Refinement Mesh Generation, Ph.D. thesis, Pittsburgh, Pennsylvania (1997).
- [12] J. R. Shewchuk, Tetrahedral Mesh Generation by Delaunay Refinement, in: *Proceedings of the fourteenth annual symposium on Computational geometry*, SCG '98, ACM, New York, NY, USA, 1998, pp. 86–95. doi:10.1145/276884.276894.  
URL <http://doi.acm.org/10.1145/276884.276894>
- [13] S. Cheng, T. Dey, Quality Meshing with Weighted Delaunay Refinement, *SIAM Journal on Computing* 33 (1) (2003) 69–93. arXiv:<http://epubs.siam.org/doi/pdf/10.1137/S0097539703418808>, doi:10.1137/S0097539703418808.  
URL <http://epubs.siam.org/doi/abs/10.1137/S0097539703418808>
- [14] S. W. Cheng, T. K. Dey, E. A. Ramos, Delaunay Refinement for Piecewise Smooth Complexes, *Discrete & Computational Geometry* 43 (1) (2010) 121–166. doi:10.1007/s00454-008-9109-3.  
URL <http://dx.doi.org/10.1007/s00454-008-9109-3>
- [15] J. D. Boissonnat, S. Oudot, Provably Good Surface Sampling and Approximation, in: *ACM International Conference Proceeding Series*, Vol. 43, 2003, pp. 9–18.
- [16] J. D. Boissonnat, S. Oudot, Provably Good Sampling and Meshing of Surfaces, *Graphical Models* 67 (5) (2005) 405–451.
- [17] C. Jamin, P. Alliez, M. Yvinec, J. D. Boissonnat, CGALmesh: A Generic Framework for Delaunay Mesh Generation, Tech. rep., INRIA (2013).
- [18] B. Delaunay, Sur la sphere vide, *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7 (793-800) (1934) 1–2.
- [19] S. W. Cheng, T. K. Dey, J. R. Shewchuk, Delaunay Mesh Generation, Taylor & Francis, New York, 2013.
- [20] H. Edelsbrunner, N. R. Shah, Triangulating Topological Spaces, *International Journal of Computational Geometry & Applications* 7 (04) (1997) 365–378.
- [21] N. Amenta, M. Bern, Surface Reconstruction by Voronoi Filtering, *Discrete & Computational Geometry* 22 (4) (1999) 481–504.
- [22] S. W. Cheng, T. K. Dey, J. A. Levine, A Practical Delaunay Meshing Algorithm for a Large Class of Domains, in: M. Brewer, D. Marcum (Eds.), *Proceedings of the 16th International Meshing Roundtable*, Springer Berlin Heidelberg, 2008, pp. 477–494. doi:10.1007/978-3-540-75103-8\_27.  
URL [http://dx.doi.org/10.1007/978-3-540-75103-8\\_27](http://dx.doi.org/10.1007/978-3-540-75103-8_27)
- [23] S. W. Cheng, T. K. Dey, E. A. Ramos, T. Ray, Sampling and Meshing a Surface with Guaranteed Topology and Geometry, *SIAM journal on computing* 37 (4) (2007) 1199–1227.
- [24] T. K. Dey, J. A. Levine, Delaunay meshing of piecewise smooth complexes without expensive predicates, *Algorithms* 2 (4) (2009) 1327–1349.
- [25] P. A. Foteinos, A. N. Chernikov, N. P. Chrisochoides, Guaranteed quality tetrahedral Delaunay meshing for medical images, *Computational Geometry: Theory and Applications* 47 (4) (2014) 539–562.
- [26] A. Bowyer, Computing Dirichlet Tessellations, *The Computer Journal* 24 (2) (1981) 162–166.
- [27] H. Erten, A. Üngör, Quality Triangulations with Locally Optimal Steiner Points, *SIAM J. Sci. Comp.* 31 (3) (2009) 2103–2130.
- [28] S. Rebay, Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer-Watson Algorithm, *Journal of Computational Physics* 106 (1) (1993) 125 – 138. doi:<http://dx.doi.org/10.1006/jcph.1993.1097>.  
URL <http://www.sciencedirect.com/science/article/pii/S0021999183710971>
- [29] D. J. Mavriplis, An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness, *Journal of Computational Physics* 117 (1) (1995) 90 – 101. doi:<http://dx.doi.org/10.1006/jcph.1995.1047>.  
URL <http://www.sciencedirect.com/science/article/pii/S0021999185710479>
- [30] P. J. Frey, H. Borouchaki, P. L. George, 3D Delaunay Mesh Generation Coupled with an Advancing-Front Approach, *Computer Methods in Applied Mechanics and Engineering* 157 (12) (1998) 115 – 131. doi:[http://dx.doi.org/10.1016/S0045-7825\(97\)00222-3](http://dx.doi.org/10.1016/S0045-7825(97)00222-3).  
URL <http://www.sciencedirect.com/science/article/pii/S0045782597002223>
- [31] J. F. Remacle, F. Henrotte, T. Carrier-Baudouin, E. Béchet, E. Marchandise, C. Geuzaine, T. Mouton, A Frontal-Delaunay Quad-Mesh Generator using the  $\mathbb{L}^\infty$ -norm, *International Journal for Numerical Methods in Engineering* 94 (5) (2013) 494–512. doi:10.1002/nme.4458.  
URL <http://dx.doi.org/10.1002/nme.4458>
- [32] D. Rypl, Sequential and Parallel Generation of Unstructured 3D Meshes, Ph.D. thesis, CTU in Prague (1998).

- [33] J. Schreiner, C. Scheidegger, C. Silva, High-Quality Extraction of Isosurfaces from Regular and Irregular Grids, *Visualization and Computer Graphics*, IEEE Transactions on 12 (5) (2006) 1205–1212. doi:10.1109/TVCG.2006.149.
- [34] A. Üngör, Off-centers: A New Type of Steiner Points for Computing Size-optimal Guaranteed-quality Delaunay Triangulations, *Computational Geometry: Theory and Applications* 42 (2) (2009) 109–118.
- [35] A. N. Chernikov, N. P. Chrisochoides, Generalized insertion region guides for Delaunay mesh refinement, *SIAM Journal on Scientific Computing* 34 (3) (2012) A1333–A1350.
- [36] P. A. Foteinos, A. N. Chernikov, N. P. Chrisochoides, Fully generalized two-dimensional constrained Delaunay mesh refinement, *SIAM Journal on Scientific Computing* 32 (5) (2010) 2659–2686.
- [37] N. Amenta, S. Choi, R. K. Kolluri, The power crust, unions of balls, and the medial axis transform, *Computational Geometry* 19 (2) (2001) 127–153.
- [38] T. K. Dey, W. Zhao, Approximating the medial axis from the Voronoi diagram with a convergence guarantee, *Algorithmica* 38 (1) (2004) 179–200.
- [39] P.-O. Persson, Mesh Size Functions for Implicit Geometries and PDE-based Gradient Limiting, *Engineering with Computers* 22 (2) (2006) 95–109. doi:10.1007/s00366-006-0014-1. URL <http://dx.doi.org/10.1007/s00366-006-0014-1>
- [40] C. Jamin, S. Pion, M. Teillaud, 3D Triangulations, in: *CGAL User and Reference Manual*, 4.6.1 Edition, CGAL Editorial Board, 2015. URL <http://doc.cgal.org/4.6.1/Manual/packages.html#PkgTriangulation3Summary>
- [41] C. Jamin, S. Pion, M. Teillaud, 3D Triangulation Data Structure, in: *CGAL User and Reference Manual*, 4.6.1 Edition, CGAL Editorial Board, 2015. URL <http://doc.cgal.org/4.6.1/Manual/packages.html#PkgTDS3Summary>